Methods to Improve the Signal Quality of Corrupted Multi-Parameter Physiological Signals

by

Gartheeban Ganeshapillai

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

Author .....
Department of Electrical Engineering and Computer Science
May 09, 2011

Certified by .....
John J. Guttag
Professor, Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Leslie. A. Kolodziejski
Chair of the Committee on Graduate Students

Methods to Improve the Signal Quality of Corrupted Multi-Parameter Physiological Signals

by

Gartheeban Ganeshapillai

Submitted to the Department of Electrical Engineering and Computer Science
on May 09, 2011, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

A modern Intensive Care Unit (ICU) has automated analysis systems that depend on continuous uninterrupted real-time monitoring of physiological signals such as Electrocardiogram (ECG), Arterial Blood Pressure (ABP), and the Photo Plethysmogram (PPG). Unfortunately, these signals are often corrupted by noise, artifacts, and missing data, which can result in a high incidence of false alarms.

We present a novel approach to improve the Signal Quality of a multi-parameter physiological signal by identifying the corrupted regions in the signal, and reconstructing them using the information available in correlated signals. The method is specifically designed to preserve the clinically most significant aspects of the signals. We use template matching to jointly segment the multi-parameter signal, morphological dissimilarity to estimate the quality of the signal segment, similarity search to find the closest match from a database of templates, and time-warping to reconstruct the corrupted segment using the matching template.

Experiments carried out on the MIT-BIH Arrhythmia Database, a multi-parameter ECG database with many clinically significant arrhythmias, demonstrate the effectiveness of the method. Our method improved the classification accuracy of the beat type by more than 700% on the signal corrupted with white Gaussian noise, and increased the similarity to the original signal, as measured by the normalized residual distance, by more than 250%. When the method was applied to the multi-parameter physiological signal data from Cinc Challenge 2010 database at Physionet.org, our method improved the classification accuracy of beat type by more than 33 times on a signal corrupted with white Gaussian noise, and increased the similarity to the original signal by more than 280%.

Thesis Supervisor: John J. Guttag
Title: Professor, Electrical Engineering and Computer Science

Acknowledgments

This thesis would not have been possible without the help and guidance of many.

As part of my graduate school application at MIT, I had to choose a professor I wanted to work with. John Guttag was not the advisor I chose, but the advisor I was assigned to. "It is often spur-of-the-moment decisions, sometimes made by others, that can change our whole lives." John Guttag's inspiration, guidance, and, more than anything, empathy have so far helped me survive the graduate school. He has been an excellent teacher, mentor, friend, and, of course, the thesis advisor. It has been a pleasure to work with John, and I hope to continue in the coming years.

I am thankful to the valuable comments from the members of the Cardiac group, especially that from Prof Collin Stultz, whose overt sarcasms and views kept the humor floating on our otherwise dry technical decisions.

I am thankful to my lab mates, Ali Shoeb - the eldest, Alaa Kharbouch, Jessica Liu, Asfandyar Qureshi, Eugene Shih, Anima Singh, Zeeshan Syed - the wisest, and Jenna Weins. Of these individuals, I would like to extend a special thanks to Alaa Kharbouch and Zeeshan Syed for helping me with the technical materials. I am also in great debt to Jessica Liu for patiently running the experiments.

This thesis would not have been possible without the dedication and kindness of Dorothy Curtis, Sheila Marian, and Janet Fischer. I owe a great debt to Dorothy Curtis. With such exuberance, she has always been a delight to converse with.

I am thankful to many friends at and around MIT. The friends I met in the first year at MIT; the friends who introduced various lifestyles of Boston; the friends I met at EECS-GSA, who made me want to attend the events at GSA, volunteer at the events, and eventually lead and organize the events for GSA. I am also indebted to fellow Sri Lankans in Boston, including the graduate students at CSAIL, who regularly help resurrect the Sri Lankan in me.

Finally, I am grateful to my parents and grandparents, who have always taken interest in my academic progress, supported me along the way. If it were not for the love of my mother, and the support of my grand parents, I would not be here.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis, we present a set of novel computational tools to improve the Signal Quality of physiological signals. We use an unsupervised learning framework that does real-time estimation of the correct values of corrupted signals by fusing information from correlated signals. There are three main components of our work: segmenting a multi-parameter quasiperiodic signal in the presence of noise and transient corruption, developing a Signal Quality Estimate that is comparable across different type of signals, and reconstructing the segments of low signal quality.

1.1 Motivation

A modern Intensive Care Unit (ICU) employs several bedside monitors to track the state of patients. They allow continuous monitoring of a patient, and inform medical staff of changes in the status of the patient [26]. Automated analysis systems are typically used to analyze these signals in real-time. They operate by triggering alarms when one or more physiological parameters fall outside the predetermined normal region [48, 52, 40].

These systems depend on continuous uninterrupted real-time monitoring of physiological signals such as electrocardiogram (ECG), arterial blood pressure (ABP), and the photo plethysmogram (PPG) [40]. Unfortunately, these signals are often corrupted by noise, artifacts, and missing data, which can result in a high incidence of

false alarms [1, 26, 6]. Frequent false alarms lead to the desensitization of clinical staff to real alarms [13]. On the other hand, in the event of missed alerts, the patient remains unattended, and the state of the patient might eventually deteriorate into serious medical condition.

It would be advantageous if an algorithm could estimate the correct values of the corrupted signals by fusing the information from correlated signals from multiple monitors. In this thesis, we address the problem of identifying the corrupted regions in a multi-parameter signal, and reconstructing them in a clinically useful way using the information available in the correlated signals.

1.2 Goal

We consider a multi-parameter signal represented by a matrix $\mathbf{S}_{n\times m}$, where each column represents a signal (e.g., ECG) and each row represents a point in time. There are m synchronous single parameter signals in $\mathbf{S}$. Each cell $s_{i,j}$ contains one sample. Samples may be corrupted in an unknown fashion. Our goal is to identify the corrupted regions, and estimate the sample values of that region.

A typical multi-parameter physiological signal consists of various ECG channels, ABP, PPG, respiration, central venous pressure (CVP), pulmonary arterial pressure (PAP) and intracranial pressure (ICP). We mainly consider ECG channels, ABP and PPG, because they are commonly available. In future we hope to extend the work to other correlated signals.

1.3 Proposed Solution

First, we identify the segment boundaries of the multi-parameter signal in the presence of significant amounts of transient corruption spanning multiple columns and rows of the matrix $\mathbf{S}$.

We use a template, a short multi-parameter signal, and match it with a sliding window of the multi-parameter signal. The initial template is derived from an

archived signal, and is regularly updated to reflect the time evolution of the signal. We continuously extract a non-overlapping window from **S**, and identify the boundary in the window by finding the prefix of the window that most closely matches the template. The matching is done using weighted time warping (WTW), which minimizes the weighted morphological dissimilarity across all the parameters. The warped distance between two single parameter signals gives the morphological dissimilarity. The weight represents the estimated quality of the signal, which is again computed by the morphological dissimilarity of the signal with its counterpart in the template. The signal quality estimated from the morphological dissimilarity is also used to find the corrupted regions.

Second, using a database of templates, we reconstruct the corrupted regions. Here, a template is a segment of the multi-parameter signal that was chosen from previously seen regions that were believed to be free of signal corruption. When we come across the segments of high signal quality, we add them to the database; thus, we learn new morphologies.

The method is based on finding the closest match (template) to the corrupted segment from the database, time-warping the template to fit the corrupted segment's interval, and replacing the corrupted segment with the result. The closest match is found using the DTW cost. As a preliminary step, we represent the segments by features. This has the dual advantages of providing a level of abstraction that preserves the clinically relevant information and speeding up the matching.

Finally, we combine the matches based on the quality of each match, and reconstruct the segment. In this process, we time-warp the matches with the target segment, so that they will be of the same length.

The proposed framework is patient specific. Here, we learn the relationships between the correlated signals for a patient, and reconstruct the corrupted regions using the history of the patient and the correlated signals. However, in future work, we would like to extend this, by sharing the databases of the same signals among the patients. Thus, we will be able to reconstruct the beats of types, previously unseen for a patient, but prevalent on others.

1.4 Problem Decomposition

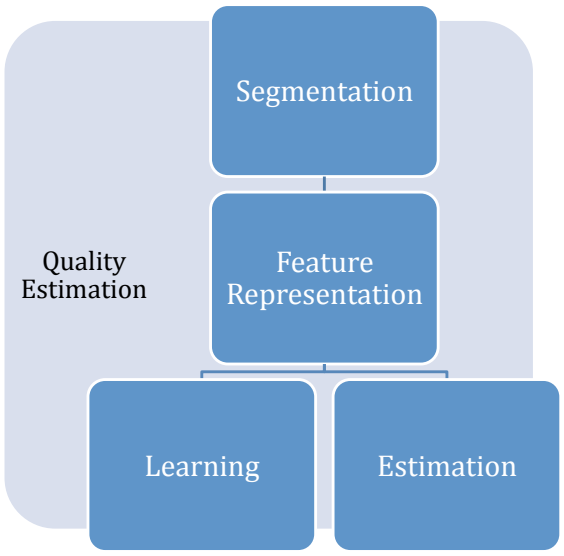


Figure 1-1: Key components of the system.

In order to modularize the framework, we structure our problem into following subproblems.

- **Segmentation** to limit the analysis to a segment (a beat) of the signal. The quasiperiodic physiological signals have a repetitive morphological structure. By restricting the focus to a segment we limit the error propagation to that segment.
- **Estimating the quality** of the signals using the change in signal morphology.
- **Feature representation** of the segments to reduce the time complexity of the comparison of two sequences and provide a level of abstraction.
- **Learning** the relationship between the correlated signals by building a database of templates.

- **Estimating** the actual values in the corrupted region using the signal's history and the correlated signals.

These subproblems depend on each other sequentially as shown in Figure 1-1. The modularity of the framework allows us to independently optimize the parts . Further, modularization is important for extending the system. The only component that depends on the signal type is the feature representation. Hence, the proposed framework can be extended to support other set of correlated quasiperiodic signals.

1.5 Contributions

We briefly discuss some of the major contributions of our work. A detailed discussion is deferred to subsequent parts of the thesis.

- We present a novel approach to **jointly segment a multiparameter signal** in the presence of significant amounts of noise and transient corruption. The key idea is that by simultaneously considering all the signals one can segment them more accurately than would be possible by considering each signal independently. Further, we formulate the problem of **segmentation as a dissimilarity minimization problem**, and exploit the repetitive morphology of the quasiperiodic physiological signals to segment them. We use template matching to segment the quasiperiodic units. We use a signal template that eliminates the need of any prior knowledge of the specific properties of the signal, and the dynamic adaptation of the template, which allows us to accommodate the time evolution of the signal.

- We propose a new method, **Weighted Time Warping (WTW)**, to measure the similarity between two multi-parameter sequences of different lengths. WTW is an extension of Dynamic Time Warping (DTW) to multiple dimensions with varying influence along each dimension. The extent of the influence exerted by each dimension is determined by the quality of the signal in that dimension.

- We propose a new method, **Morphological dissimilarity (MD)** to estimate the signal quality at the granularity of a beat. MD is comparable across different types of signals, and is therefore, useful to obtain the relative signal qualities across multiple signals in a multi-parameter signal.

- We propose a new method to **reconstruct the corrupted segments** of a signal, by matching the corresponding segment from the correlated signal with a set of templates in a database.

- We propose a new method for **similarity search on a variable length physiological time series data** by using features. This serves dual purposes. First, it provides a level of abstraction and thus generalizes. Second, it reduces the computational complexity of the database lookups for a variable length signal.

1.6 Organization of Thesis

The remainder of this thesis is organized as follows. Chapter 2 presents the previous work and background. Chapter 3 introduces the problem of segmenting a quasiperiodic multi-parameter signal in the presence of corruption. Chapter 4 presents a method to estimate the signal qualities of different types of quasiperiodic physiological signals. Chapter 5 describes the methods to identify the corrupted segments and reconstruct them. In Chapter 6, we discuss the fusion of multiple signals and the regression on the global trends. Chapter 7 details the implementation of the system. Chapter 8 concludes with a summary and discussion of future work.

Chapter 2

Background

In this chapter, we review the related clinical applications, and provide the technical background for our work. In Section 2.1, we start with a short discussion on the attempts to counteract noise and transient corruption on physiological signals by fusing data from multiple sources. In Section 2.2, we discuss the related work, especially the methods submitted to Mind the Gap contest. We devote the remainder of the chapter for the discussion on the technical material related to our work. In Section 2.3, we present the existing methods for signal quality estimation. In Section 2.4, we discuss the characteristics of different distance functions. In Section 2.5, we look at different methods for efficient lookups on multi-dimensional data.

2.1 Redundancy and Data fusion

Corruptions by noise, artifact, and missing data in physiological signals lead to serious errors in automated medical systems and early warning systems. Researchers have proposed several methods to counteract this. A survey of strategies that address this problem is provided in [38]. Recent attempts to mitigate this problem focus on using redundant measurements, and fusing data from multiple sensors [1, 13, 38, 12]. Typically, they employ independent methods on different channels and combine the results only at the final stage.

Recently there have been attempts to robustly estimate heart rate (HR) by fusing

information from multiple signals [38]. In addition to various ECG channels, researchers also make use of Arterial Blood Pressure (ABP) and Photo Plethysmogram (PPG) signals for HR estimation [12]. In this context, segmentation methods for ABP and PPG by detecting the parts of the waveform corresponding to the onset of a pulse have been proposed [1, 13]. These methods independently segment the physiological signals. Hence, they do not preserve the alignment between the segments across different signals.

2.2 Filling the Gap

The problem of reconstructing a corrupted multi-parameter physiological signal was formally posed in the 11^{th} annual PhysioNet/CinC challenge - Mind the Gap. Contestants were asked to reconstruct, using any combination of available prior and concurrent information, 30-second segments of ECG, continuous blood pressure waveforms, respiration, and other signals that had been removed from recordings of patients in intensive care units. The data collected from MIMIC II project was used for the contest. The submissions were rated by two criteria Q_1 and Q_2.

$$V_{res}(t) = V_{rec}(t) - V_{ref}(t), \quad \text{where } t = t_0, t_0 + \Delta t, ..., (n-1)\Delta t. \tag{2.1}$$

$$E_{ref} = \sum_{i=0}^{n} V_{ref}(t_0 + i\Delta t)^2 - \frac{1}{n}(\sum_{i=0}^{n} V_{ref}(t_0 + i\Delta t))^2 \tag{2.2}$$

$$Q_1 = max(1 - \sum_{i=0}^{n} V_{res}(t_0 + i\Delta t)^2 / E_{ref}, 0) \tag{2.3}$$

$$Q_2 = cov(V_{ref}(t), V_{rec}(t)) / \sigma_{V_{ref}} \sigma_{V_{rec}} \tag{2.4}$$

Here, the target signal $V_{ref}(t)$, is subtracted from the reconstruction, $V_{rec}(t)$ to obtain the residual signal, $V_{res}(t)$. E_{ref} is the energy (sample variance) of the target signal [41].

In the decreasing order of the scores earned, the approaches of some of the more successful contestants are discussed below.

The highest scored method [47] uses a multilayer perceptron neural network

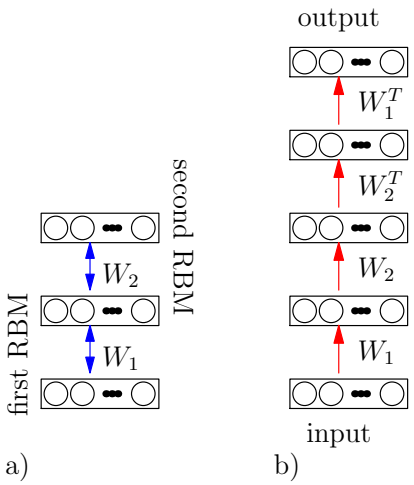


Figure 2-1: An autoencoder (b) is built using two RBMs (a) stacked together [47].

(MLP) that outputs the signal to be reconstructed, the target signal. The MLP maps all the channels present to the channel to be reconstructed. The neural network has four hidden layers, and uses the ideas from [30] to train the network. An auto-encoder is used for MLP whose output layer reproduces the input (Figure 2-1). The middle layer acts as a code to represent the input. The layers until the middle layer define an encoder, and the layers after the middle layer, inclusive, form a decoder. Two Restricted Boltzmann Machines (RBM) are stacked to built the auto-encoder.

Silva [49] produces M reconstruction signals from each of the M channels using a bank of GALL filters [17]. The signals are then linearly combined using a Kalman filter. The weights of the Kalman filter are learned in the training phase, and they remain constant over the remainder of the signal.

Similarly, another method [25] that uses a bank of adaptive filter, maps each of the M channels to the channel that needs to be reconstructed, and averages the output (Figure 2-2).

Li *et al.* [39] proposes to reconstruct the corrupted signals by matching patterns. The paper uses a combination of correlation coefficient and Euclidean distance to find the match, a template from the signal's history, and replaces the corrupted region with the template. Since the region of reconstruction is likely to be of different length compared to the length of the template, the authors fill the gaps with 0s, or trim the

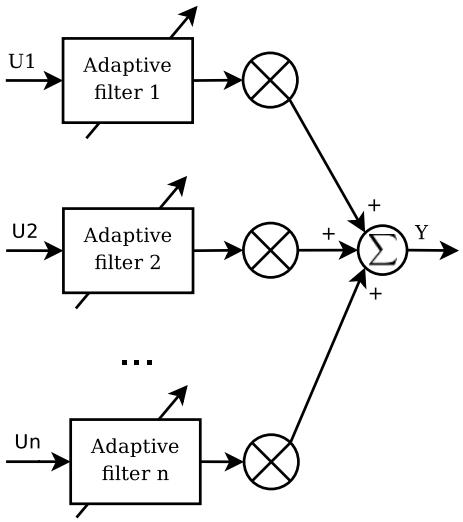


Figure 2-2: Input channels U_i are passed through adaptive filters, and summed up. The adaptive filter coefficients are estimated from the negative feedback through a closed loop [25].

trailing signal.

In the contest, these methods were evaluated using the residual distances on a set of multi-parameter database containing ECG, ABP, and PPG signals. Table 2.1 summarizes the performances reported by these methods.

Table 2.1: Reported reconstruction performances of the methods submitted to PhysioNet/CinC challenge 2010.

Entrant	Q_1	Q_2
Neural newtork [47]	83.00	90.51
Gall Filter + Kalman Filter [49]	70.55	84.09
Adaptive Filter + Averaging [25]	69.66	81.32

2.3 Signal Quality Estimation

Our framework requires the assessment of the signal quality at multiple stages of the system. In the joint segmentation of the multi-parameter signals (Chapter 3), we weigh the individual signal's influence on the joint segmentation of the multi-parameter signal by the estimate of the quality of the signal. In the reconstruction of

the morphological features of the physiological signals (Chapter 5), we use the signal quality estimates to identify the presence of noise and corruption in the signal.

Researchers have investigated the signal quality assessments of many physiological signals including ECG, ABP and PPG [11, 38, 44, 63]. These papers use the performance of the QRS detectors that rely on distinct properties of the ECG signal, and the spectral power characteristics of the ECG, ABP and PPG signals to estimate the signal quality. More recently, researchers have started to look into the morphological dissimilarity to identify the presence of the artifacts [22, 57].

2.3.1 ECG Signal Quality

Signal quality assessment for a single channel ECG is presented in [57]. It estimates the signal quality by computing the area differences (mismatch) between successive QRS complexes. A cumulative histogram of mismatches is then generated, and the signal quality is determined based on how fast the cumulative histogram curves rise.

Signal quality assessment for a multi-lead ECG is presented in [38]. It combines the following four distinct methods to obtain an ECG signal quality metric.

bSQI

This metric is based on the hypothesis that the different QRS detection algorithms are sensitive to different types of noise. For a window of length w seconds, it derives the quality estimates from the ratio of the beats detected synchronously by two different algorithms to all the detected beats [44].

$$bSQI(k) = N_{matched}(k, w)/N_{all}(k, w) \tag{2.5}$$

The window length w is set to 10 seconds in this and the following equations [38]. N_{matched} is the numbers of beats both methods agreed upon and N_{all} is the number of all beats detected by either method. The method uses a digital filtering and integration (DF) based QRS detector [24], and a length transform (LT) based QRS detector [62]. Here, LT is more sensitive to noise than DF.

iSQI

This measures the inter-channel signal quality using the ratio of the number of matched beats to the number of all detected beats between a given lead i and all other synchronous ECG leads. While other constraints remain the same as for bSQI, iSQI uses only DF.

$$iSQI(k)_i = max\{N_{matched}(k,w)_{i,j}/N_{all}(k,w)_{i,j}\}, \forall j, j \neq i \tag{2.6}$$

kSQI

This metric is based on the kurtosis of the ECG signal. It works on the premise that the samples from random noise is uncorrelated and tends to have a Gaussian distribution, and the samples from a signal such as ECG are correlated and tend to have non-Gaussian distributions. Kurtosis measures the related peakedness of a distribution with respect to a Gaussian distribution.

$$kSQI(k)_i = \begin{cases} 1 & \text{if } kurtosis(k,w) > 5 \\ 0 & \text{if } kurtosis(k,w) \leq 5 \end{cases} \tag{2.7}$$

sSQI

sSQI uses Power Spectral Density (PSD) distribution to determine the quality of the signals. For ECG, QRS energy is centered around 10 Hz. sSQI compares the PSD in this region to the whole signal. If P gives the signal power, Spectral Distribution Ratio (SDR), and sSQI (from SDR) are calculated by,

$$SDR(k) = \int_{f=5}^{f=14} P(k,w) \Big/ \int_{f=5}^{f=50} P(k,w) \tag{2.8}$$

$$sSQI(k) = \begin{cases} 1 & \text{if } SDR(k) > 0.5 \text{ and } SDR(k) \leq 0.8 \\ 0 & \text{if } SDR(k) < 0.5 \text{ or } SDR(k) > 0.8 \end{cases} \tag{2.9}$$

ECGSQI

The combined signal quality index (ECGSQI) is then given by

$$ECGSQI(k) = \begin{cases} max(bSQI(k), iSQI(k)) & \text{if } kSQI(k) = 1 \ \& \ sSQI(k) = 1 \\ max(bSQI(k), iSQI(k)) * \eta & \text{if } kSQI(k) = 0 \ \& \ sSQI(k) = 1 \\ bSQI(k) & \text{if } kSQI(k) = 1 \ \& \ sSQI(k) = 0 \\ bSQI(k) * \eta & \text{if } kSQI(k) = 0 \ \& \ sSQI(k) = 0 \end{cases} \tag{2.10}$$

where $\eta = 0.7$. Each metric contributes to a lowered overall ECGSQI during the noisy period.

2.3.2 ABP Signal Quality

Li *et al.* [38] combines wSQI and jSQI to obtain the arterial blood pressure signal quality (ABPSQI). The wSQI algorithm uses ABP pulse detection and fuzzy reasoning to estimate the signal quality [63]. The wSQI takes a continuous value between zero and one (poor to excellent quality) for each beat. The jSQI algorithm uses the same ABP pulse detection routine, and series of features such as ABP amplitudes, slopes, and beat-to-beat variations in each pulse to generate the abnormality index, jSQI. It takes 0 for normal beats, and 1 for physiologically abnormal beats or noise artifacts [51].

ABSQI is given by,

$$ABPSQI(k) = \begin{cases} wSQI(k) & \text{if } jSQI(k) = 0 \\ wSQI(k) * \eta & \text{if } jSQI(k) = 1. \end{cases} \tag{2.11}$$

where $\eta = 0.7$.

2.3.3 PPG Signal Quality

Gil *et al.* [18] identifies the periods of major artifact in photo plethysmogram by thresholding the Hjorth parameters of the waveform. The Hjorth parameters represent a time signal by its activity (H_0), mobility (H_1), and complexity (H_2) [31]. To obtain the corruption free PPG signal, Deshmane [13] proposes to threshold H_1 and H_2 values calculated on a 5 s window of the PPG signal, between $\tilde{H}_1 - 1$ and $\tilde{H}_1 + 1.4$, and below $\tilde{H}_2 + 3$ respectively. Here $\tilde{H}_1$ and $\tilde{H}_2$ are the medians of the H_1 and H_2 for the whole signal. A detailed discussion on the computation of the Hjorth parameters and PPG signal quality estimation are given in [13].

2.4 Distance Functions

The critical component of time series analysis is the choice of the distance function. A distance function is often used to capture the notion of similarity between two sequences in these analyses [42].

We use distance functions extensively in our methods. For instance, in Chapter 3, we use template matching for the segmentation, where we minimize the distance between the template and a window. In Chapter 4, we use distance function to estimate the morphological dissimilarity, and thus quantify the signal quality. In Chapter 5, we obtain the template closest to the current segment from the database using a distance function.

A diverse set of measures exists for estimating the distance between two sequences. They can be divided into several categories. Elasticity is a criterion that can be used to categorize these measures. Euclidean distance is a non-elastic distance measure, where as, Dynamic Time Warping (DTW), Longest Common Subsequence (LCSS), and Edit Distance on Real Sequence (EDR) are elastic measures. Elastic measures are useful to compare two sequences of different lengths by time-warping the sequences. The resulting extent of warping is usually dependent on the sequence, and is often exacerbated by the presence of noise. Another categorization is based on the objective functions of the measures. Euclidean distance and DTW are based on L_2 norm. LCSS

and EDR compute a similarity score based on a matching threshold ϵ. All of the elastic distance measures rely on dynamic programming for their efficient evaluation. Nevertheless, typical time-space complexity of these methods is quadratic on the lengths of the sequences. To address these problems, a number of techniques have been developed that impose restrictions on the warping length of the dynamic programming evaluation [33, 43, 42, 56].

2.4.1 Euclidean Distance

Euclidean distance is the most basic distance measure. It sums the pair-wise distances between the points in two time series. Given two time series, A and B of length n, Euclidean distance is given by $\sqrt{\sum_{i=1}^{n}(A_i - B_i)^2}$. It is extended to higher dimensions by taking the L_p norm of the pair wise distances as in $\sqrt[p]{\sum_{i=1}^{n}|A_i - B_i|^p}$. For example, for two dimensional sequences $A \in \Re^{n\times 2}$ and $B \in \Re^{n\times 2}$, under L_2 norm, Euclidean distance is given by $\sqrt{\sum_{i=1}^{n}(A_{1,i} - B_{1,i})^2 + (A_{2,i} - B_{2,i})^2}$. This measure can be used to compare only two sequences of equal length. More sophisticated techniques that employ time-warping are therefore needed to compare two sequences of different lengths [14].

2.4.2 Dynamic Time Warping

Given two time series, A and B of length n and m, DTW finds the optimal alignment between A and B. It achieves this by minimizing the warped distance between A and B.

The element wise distance between two points from A and B is given by $c_{(x,y)} = (A_x - B_y)^2$. We construct the pairwise distance matrix by applying this process between all pairs of elements between A and B. The optimal alignment $W = w_1 w_2..w_k$ is found by choosing the pairs incrementally from $(1, 1)$ to (n, m) such that the cumulative distance of the chosen pairs is minimized. Here, k is the alignment length.

Dynamic programming is used to compute the cumulative distance efficiently and the resulting matrix is called the accumulated distance matrix. An element $d_{x,y}$ in the

accumulated distance matrix gives the minimum distance between two sub-sequences $A_x = a_1 a_2..a_x$ and $B_y = b_1 b_2..b_y$. Further, $d_{m,n}$ gives the minimal distance between the complete sequences A and B. Dynamic programming reduces the time complexity from exponential to $O(nm)$. Dynamic programming employs the following recursion.

$$d_{x,y} = c_{x,y} + min\{d_{x-1,y-1}, d_{x,y-1}, d_{x-1,y}\}. \tag{2.12}$$

The following three fundamental conditions constrain the computation of DTW.

- **Continuity:** In DTW, we ensure that no element from either sequence is skipped in the alignment. Hence, two consecutive points $w_t = (x, y)$ and $w_{t+1} = (x', y')$ in an alignment can only vary by at most one element in each sequence, i.e., $(x - x') = 0$ or 1, and $(y - y') = 0$ or 1.

- **Monotonicity:** In DTW, we incrementally choose the pairs for the alignment. For instance, after aligning a_x with b_y, we can only match $a_{x'}$ for $x' \geq x$ with $b_{y'}$ for $y' \geq y$.

- **Boundary conditions:** In DTW, we ensure that $w_1 = (1, 1)$ and $w_k = (n, m)$. The alignment should start at the first elements of the sequences and end at the last elements of the sequences.

Applications

Researchers are increasingly using Dynamic Time Warping (DTW) for temporal segmentation problems [35, 61]. DTW has also been used in locating motion clips [20] and temporal segmentation of human motions [61]. Vlachos *et al.* [56] provides an overview on the implementation of multidimensional Dynamic Time Warping (DTW) over the L_p norm. Based on this work, we propose weighted time warping (WTW), a novel method that uses time warping to perform segmentation in Section 3.3.2.

Limitations

In its general form, DTW is extremely flexible in matching two warped sequences. Hypothetically, in an extreme case, Equation 2.12 allows all but the last point of sequence A to be matched with the first point of B, while the last point of A is matched with all but the first point of B. This flexibility is generally beneficial. However, under noise, it can result in degenerative matchings (Figure 2-3). This has prompted researchers to look for additional constraints on or alternatives to DTW [32].

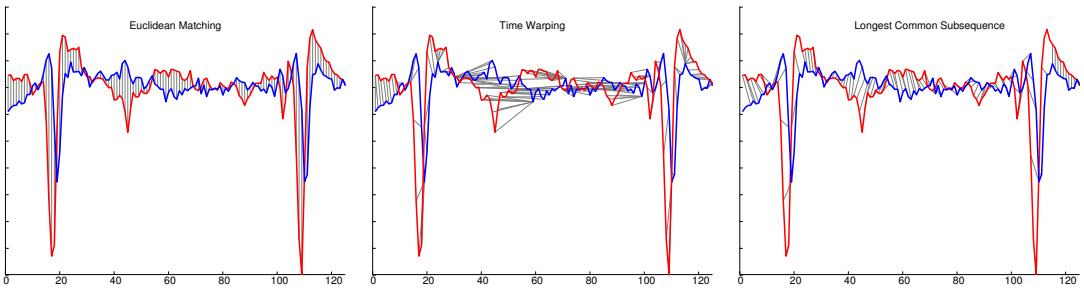


Figure 2-3: A comparison of alignments induced by a) Euclidean distance b) Time warping and c) Longest common subsequence. Euclidean matching doesn't degenerate, but incurs the highest cost (total area of difference). Time Warping achieves the minimum cost, but degenerates. Degeneration is noticeable between time 40-60. Longest common subsequence, an alternative to time warping, though not optimal, minimizes the cost, and avoids the degenerative matchings [56].

Alternatives such as LCSS and MLM have been proposed to address this issue at the expense of the flexibility offered by DTW [37, 56]. For acoustic signal processing, Myers *et al.* [43] proposes a set of local continuity and global path constraints (Figure 2-4) . These constraints help avoid degenerative matching while retaining the useful properties of the DTW.

Derivative Dynamic Time Warping

DTW finds the alignment between two sequences by minimizing the global time-warped distances between two sequences. Therefore, it fails to match the local vari-

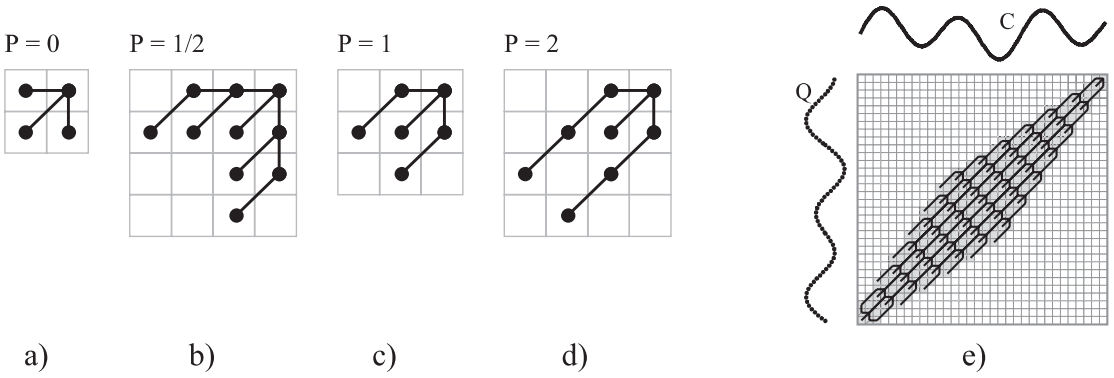


Figure 2-4: Four different local constraints corresponding to the following recursions
a) $d_{(x,y)} = c_{(x,y)} + min\{d_{(x-1,y-1)}, d_{(x,y-1)}, d_{(x-1,y)}\}$
b) $d_{(x,y)} = c_{(x,y)} + min\{d_{(x-1,y-1)}, d_{(x-1,y-2)} + c_{(x,y-1)}, d_{(x-2,y-1)} + c_{(x-1,y)}, d_{(x-1,y-3)} + c_{(x,y-1)} + c_{(x,y-2)}, d_{(x-1,y-3)} + c_{(x,y-1)} + c_{(x,y-2)}\}$
c) $d_{(x,y)} = c_{(x,y)} + min\{d_{(x-1,y-1)}, d_{(x-1,y-2)} + c_{(x,y-1)}, d_{(x-2,y-1)} + c_{(x-1,y)}\}$
d) $d_{(x,y)} = c_{(x,y)} + min\{d_{(x-1,y-1)}, d_{(x-2,y-3)} + c_{(x,y-1)} + c_{(x-1,y-2)}, d_{(x-3,y-2)} + c_{(x-1,y)} + c_{(x-2,y-1)}\}$ and
e) the resulting global path constraint resulting by the local continuity constraint given in d). [32]

ations. Derivative Dynamic Time Warping (DDTW) [33] addresses this particular shortcoming of the DTW in computing the warped distance.

The reasoning behind this is that, if two sequences have variabilities in Y-axis, the DTW is likely to produce less desirable alignments. If those variabilities are global differences, such as offsets, scalings or linear trends, they can be removed in preprocessing. However, if the differences are local it fails to produce the correct alignment. For instance, when one local feature (e.g. valley) is more prominent in one sequence than in the other sequence, this would result in an extreme warping to minimize the global distances between two sequences that DTW is trying to optimize. DTW only considers the data-points in Y-axis. In contrast, DDTW considers the first derivative, which include the changes (variability) in the Y-axis. For instance, in Figure 2-5, DTW is unable to align the local maxima, where as DDTW provides the desired alignment [33].

In DDTW, we obtain the derivatives of the sequences, and compute the warped

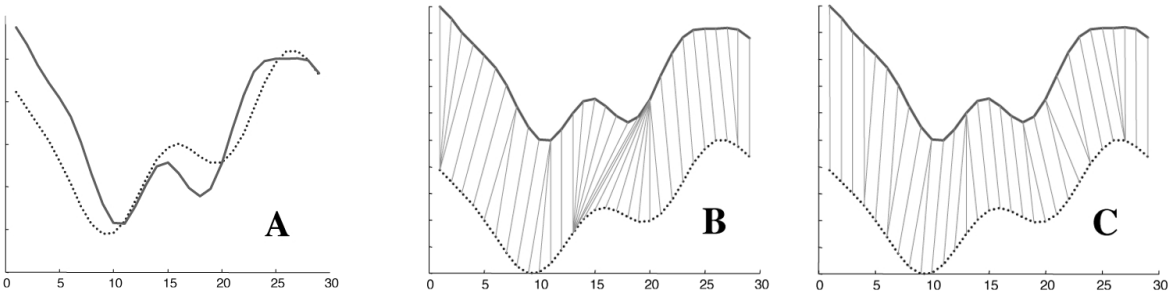


Figure 2-5: Two Sequences (A) are matched using DTW (B) and DDTW (C). DTW tries to match the peaks with peaks and valleys with valleys by extremely warping the X-axis. Further, the alignment is obtained by globally minimizing the warped distance between two sequences, which often result in degenerative matchings. For example, in (B), one peak is ultimately matched with a valley. In (C), DDTW matches a local maxima with a local maxima, and hence minimizes the warped distance locally. [33]

distance on the derivatives.

$$A'_t = \frac{(A_t - A_{t-1}) + (A_{t+1} - A_{t-1})/2}{2} \qquad 1 < t < n \tag{2.13}$$

$$B'_t = \frac{(B_t - B_{t-1}) + (B_{t+1} - B_{t-1})/2}{2} \qquad 1 < t < m \tag{2.14}$$

$$c_{i,j} = (A'_i - B'_j)^2 \tag{2.15}$$

$$d = \begin{bmatrix} c_{1,1} & c_{1,2} & .. & c_{1,m} \\ c_{2,1} & c_{2,2} & .. & c_{2,m|} \\ ... & .. & .. & .. \\ c_{n,1} & .. & .. & c_{m,n} \end{bmatrix} \tag{2.16}$$

Here $A \in \Re^n$ and $B \in \Re^m$ are two sequences, and we want to compute the alignment between them; A' and B' are the derivatives of the original sequences.

2.4.3 Longest Common Sub Sequence

LCSS is a variation of the edit distance that is extended to real sequences. It allows two sequences to be stretched for matching without rearranging the order of the elements by allowing some elements to be unmatched. While DTW requires all the elements, including the outliers, to be matched, LCSS allows few elements to be

unmatched. This gives the method the resilience against the noise [56].

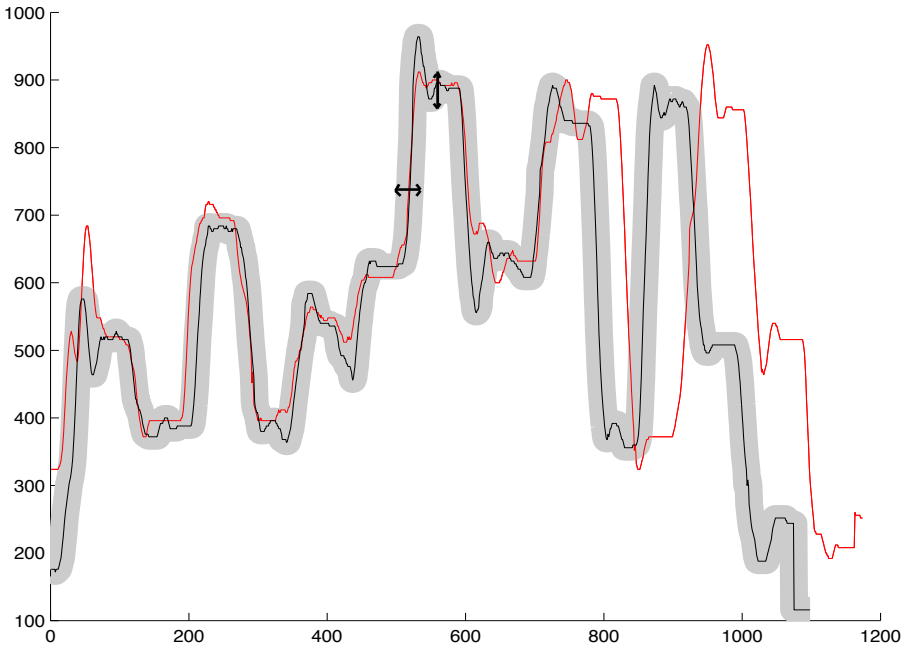


Figure 2-6: The common subsequences for two sequences as identified by LCSS. The gray region defines the area that falls within the thresholds δ_x and δ_y of the first sequence. The similarity is maximized by finding the longest sub-sequences from both sequences that share the common area [56].

Given two time series, A and B of length n and m, LCSS maximizes the similarity between A and B by finding the longest subsequences from both sequences whose elements fall within δ_x and δ_y of each other in X-axis and Y-axis respectively (Figure 2-6). The following dynamic programming function on two sub-sequences $A_x = a_1 a_2..a_x$ and $B_y = b_1 b_2..b_y$ is used to achieve this [56].

$$LCSS(A_x, B_y) = \begin{cases} 0 & \text{if A or B is empty} \\ 1 + LCSS(A_{x-1}, B_{y-1}) & \text{if } |a_x - b_y| < \delta_y \text{ and} \\ max\{LCSS(A_{x-1}, B_y), & |x - y| < \delta_x \\ LCSS(A_x, B_{y-1})\} & \text{otherwise} \end{cases} \tag{2.17}$$

Threshold δ_x controls the time-warping permitted and δ_y bounds the pair wise differences of two similar elements. Finally, the similarity (S) and the distance (D) between

two sequences A and B are obtained from LCSS by

$$S(A,B) = \frac{LCSS(A,B)}{min(n,m)} \tag{2.18}$$

$$D(A,B) = 1 - S(A,B) \tag{2.19}$$

LCSS is extended to higher dimensions by maximizing the common volume, instead of the common area [56]. For two sub-sequences $\mathbf{A}_x = a_1 a_2..a_x$ and $\mathbf{B}_y = b_1 b_2..b_y$ of $\mathbf{A} \in \Re^{n \times d}$ and $\mathbf{B} \in \Re^{m \times d}$, the resulting dynamic programming function is

$$LCSS(\mathbf{A}_x, \mathbf{B}_y) = \begin{cases} 0 & \text{if } \mathbf{A} \text{ or } \mathbf{B} \text{ is empty} \\ 1 + LCSS(\mathbf{A}_{x-1}, \mathbf{B}_{y-1}) & \text{if } |a_{x,j} - b_{y,j}| < \delta_y \forall j = 1, 2..d \\ & \text{and } |x - y| < \delta_x \\ max\{LCSS(\mathbf{A}_{x-1}, \mathbf{B}_y), & \\ LCSS(\mathbf{A}_x, \mathbf{B}_{y-1})\} & \text{otherwise} \end{cases} \tag{2.20}$$

LCSS, like other edit distance based measures, is highly resilient to noise. Further, not only it lets us control the extent of warping by δ_x, it also allows us to decide the pair wise similarity between two elements in the sequences by δ_y. Owing to these properties, we choose LCSS for generating the morphological dissimilarity in Section 4.3, which in return estimates the signal quality.

2.4.4 Edit Distance on Real Sequence

EDR is based on Edit Distance (ED), which is used on strings to measure the similarity. For two strings A and B, ED measures the number of edits (insert, delete or replace operations) required to convert A into B. EDR extends this to real sequences, by quantifying the match between two real values and defining the edit operations [9].

Given two time series, A and B of length n and m, EDR maximizes the similarity between A and B by finding the minimal number of edits required to convert A into B. The following dynamic programming function on two sub-sequences $A_x = a_1 a_2..a_x$

and $B_y = b_1 b_2 .. b_y$ is used to achieve this [9].

$$EDR(A_x, B_y) = \begin{cases} x & \text{if y = 0} \\ y & \text{if x = 0} \\ min\{EDR(A_{x-1}, B_{y-1}) + subcost & \text{otherwise} \\ EDR(A_{x-1}, B_y) + 1, & \\ EDR(A_x, B_{y-1}) + 1\} & \end{cases} \quad (2.21)$$

$$subcost = \begin{cases} 0 & \text{if } |a_x - b_y| < \delta \\ 1 & \text{otherwise} \end{cases} \quad (2.22)$$

Here, the costs of all edit distances are assumed to be one, and the threshold δ bounds the pair wise similarity between two elements in the sequences. Similar to LCSS, EDR can be extended to higher dimensions as well.

2.4.5 Comparison

DTW and its variant DDTW, LCSS, and EDR support elastic matching between two sequences of different lengths. While providing this feature, they incur quadratic computational complexities. There have been constraints proposed to reduce this computational complexity. Since these constraints usually limit the time-warping permitted, their use often results in improved performances [15].

Table 2.2: Distance Functions

Measure	Time Shifting	Noise	X-threshold	Y-threshold	Computational Cost
Euclidean					$O(n)$
DTW	✔				$O(n^2)$
LCSS	✔	✔	✔	✔	$O(n^2)$
EDR	✔	✔		✔	$O(n^2)$

Table 2.2 summarizes the characteristics of these distance functions [9]. DTW has the highest classification accuracy (finds the most similar sequence), and the least

resilience against noise [15]. LCSS, like other edit distance based measures, offers the best performance against the noise. EDR, while offering the noise performance similar to LCSS, achieves the classification accuracy close to DTW [15]. We use DTW, LCSS, and their extensions at different places in our work.

2.5 Similarity Search

Lookup for the closest match on a database of time series data is addressed by the Nearest Neighbor Search (NNS) optimization problem that is expressed in the metric space. The problem can be formulated as : given a set S of samples in a metric space M, to the query $q \in M$, find the closet sample in S.

The closest match can be simply found by exhaustively traversing through the database. This naive approach has the time complexity of $O(nc)$, where n is the size of the database, and c is the computational complexity of the comparison of any two entries [58]. For an elastic distant function such as DTW, LCSS, or EDR, c is proportional to d^2, where d is the dimensionality of the entries. There exist methods for lookups on a database in the metric space with the guaranteed time bounds. The two popular methods are partitioning into trees, and hashing into buckets. Locality Sensitive Hashing (LSH), Grid, Quadtree, R-tree, PR-tree, and kd-tree are few methods addressing this problem [10, 28, 29, 58].

2.6 Summary

In this chapter, we reviewed the related clinical applications. We discussed the existing signal quality metrics, and reviewed a set of distance functions. We also looked at few related tools and methods. With this context in place, we now present our work on developing novel methods to identify the corrupted regions on a multi-parameter physiological signal. In subsequent chapters, we show that our work can improve the accuracy of the classification tasks on these physiological signals in the presence of corruption.

Chapter 3

Temporal Segmentation of Multi-parameter Physiological Signals

This chapter discusses the development of a method to segment the physiological signals in the presence of noise and transient corruption. Specifically, we detail the solution to the first subproblem, the segmentation of a multi-parameter physiological signal, proposed in Section 1.4. We introduce the problem in Section 3.1 and detail the preliminaries. In Section 3.2, we present the background. We begin Section 3.3 with an overview of the algorithm followed by a detailed discussion on the subcomponents of the algorithm. Section 3.4 gives the details of the experiments carried out, and the results. Finally, we conclude this chapter with the discussion and summary.

3.1 Introduction

We address the problem of online segmentation of a quasiperiodic multi-parameter physiological signal in the presence of noise and transient corruption.

We consider a multi-parameter signal represented by a matrix $\mathbf{S}_{n\times m}$, where each column represents a signal (e.g., ECG) and each row represents a point in time. There are m synchronous single parameter signals in $\mathbf{S}$. Each cell $s_{i,j}$ contains one sample.

For simplicity, we assume that all the signals are sampled at the same rate. Because the matrix $\mathbf{S}$ represents a quasi-periodic multi-parameter signal, it has a repetitive structure that is shared by all the columns. This is common in situations, where each signal is generated by the same underlying system. In the case of the cardiovascular system, the periodicity of the blood pressure waveforms, and the plethysmograph signals are related to the heart rate.

The goal is to segment $\mathbf{S}$ in to quasiperiodic units $\mathbf{Y}$. However, in practice, samples may be corrupted in an unknown fashion. We present a novel approach to identifying segment boundaries in the presence of significant amounts of transient corruption spanning multiple columns and rows of the matrix $\mathbf{S}$. The key idea is that, by simultaneously considering all the signals one can segment them more accurately than would be possible by considering each signal independently.

In our method, we use a template $\mathbf{Z}_{\ell \times m}$ (Figure 3-1), a short multi-parameter signal that is regularly updated based upon recent estimates. The template length ℓ is roughly twice the length of the estimate of current segment. The initial template is derived from an archived signal.

The method is based on matching a sliding window $\mathbf{W}_{(\ell+e)\times m}$ of the signal to the template. Here, e is some buffer length. We continuously extract non-overlapping windows from $\mathbf{S}$ and attempt to identify the boundary in the window, finding the prefix of the window that most closely matches the template. The matching is done using a new method, weighted time warping (WTW) that minimizes the weighted morphological dissimilarity across all the parameters. The warped distance between two single parameter signals gives the morphological dissimilarity. The weight represents the estimated quality of the signal.

The method can accurately segment a multi-parameter signal, even when all the individual channels are so corrupted that they cannot be individually segmented. Further, our method accommodates the evolution of the signals by updating the template regularly.

3.2 Background

The analysis of the physiological signals typically begins with the temporal segmentation of these signals in order to modularize the process. The difficulty of segmenting non-stationary signals such as EEG and acoustic signals generally depends prior knowledge of the stationary structure of the kind of the signal. Whereas, the segmentation of quasi-periodic signals such as ECG generally makes use of the prior knowledge (e.g, QRS complex) [38].

QRS detection is widely used for the segmentation of ECG signals. Further, it is used for Heart Rate (HR) estimation, and as an entry point for classification of the types of cardiac cycles [12]. Kohler [34] provides an extensive survey on different software QRS detection methods. In the presence of noise and transient corruption, the prior knowledge will be invalid, and the accuracy of these methods deteriorates significantly. Therefore, the use of other correlated signals has been proposed [1, 12, 13, 38].

3.3 Method

First, we present the overview of the algorithm, followed by the discussion of a single iteration of the loop. Additional details are provided later in this section. Table 3.1 lists the variables introduced in the description of the algorithm. Bold capital letters represent a matrix $\mathbf{S}$, non-bold capital letters denote a column vector A, and lower-case letters denote a scalar w.

3.3.1 Overview

Goal : Let $\mathbf{S} \in \Re^{n \times m}$ be a multi-parameter timeseries consisting of set of m number of single parameter physiological signal, and $\mathbf{Z}_{\ell \times m} = \{Z_j \in \Re^{\ell}\}_m$ an initial template. The goal is to segment $\mathbf{S}$ into a set of quasiperiodic units $\mathbf{Y} = \{\mathbf{Y}_i\}$ where $\mathbf{Y}_i \stackrel{def}{=} \mathbf{S}_{[p_i, p_{i.+1})}$, and each unit corresponds to a single heart beat. Here, $\mathbf{S}_{[p_i, p_j)}$ denotes the window in the target sequence $\mathbf{S}$ from time $t = p_i$ to $t = p_j - 1$.

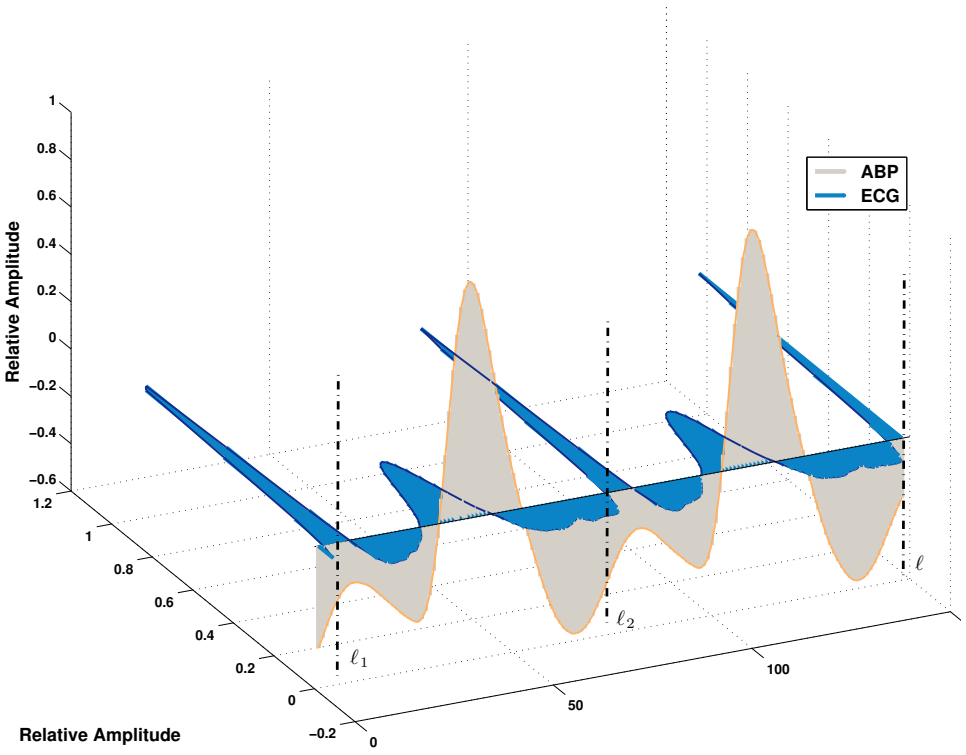


Figure 3-1: An example template. It contains clean ECG and Arterial Blood Pressure (ABP) waveforms. The positions of the segment boundaries are denoted by ℓ_1, ℓ_2 and ℓ. The template is a little longer than two segments. It contains two full segments of length $\ell_2 - \ell_1$ and $\ell - \ell_2$; the length of the template is ℓ.

Table 3.1: The summary of the variables used in the algorithm. The first row contains the variable at the initialization of the algorithm. The second row contains the list of variables used in an iteration. The last row contains the output variable.

n	Length of the multi-parameter signal.
m	Number of parameters in the multi-dimensional signal.
$\mathbf{S}$	The multi-dimensional signal that is to be segmented. It is of size $n \times m$. It contains m single parameter signals.
$\mathbf{Z}$	The initial template (a short multi-parameter signal). It is of size $l \times m$. It contains m single parameter signal templates.
p_{start}	An arbitrary starting point of the iteration.
p_i	The start of the segment at the i^{th} iteration.
ℓ	Length of the current template.
e	Buffer size.
$\mathbf{W}$	The window extracted from $\mathbf{S}$. It is of size $(\ell + e) \times m$. It contains m single parameter signal windows.
$\mathbf{pD}^j$	The pairwise distance matrix between W_j and Z_j. The cell (x, y) contains $(W_{x,j} - Z_{y,j})^2$.
q_j	Quality of the signal.
$\mathbf{D}$	Distance matrix produced by WTW.
$\mathbf{aD}$	Accumulated distance matrix obtained by applying dynamic programming on $\mathbf{D}$.
$\mathbf{W}*$	Prefix of the window $\mathbf{W}$ that matches the template $\mathbf{Z}$.
$\mathbf{W}.f*$	The row in the window $\mathbf{W}*$ that is aligned with the row in the template $\mathbf{Z}.\ell_2$.
p_{i+1}	The end of the segment at the i^{th} iteration.
$\mathbf{Y}$	Output : segments (set of quasi-periodic units).

We require the template (Figure 3-1) to be comprised of at least two segments. These segments are used to find the quasiperiodic unit $\mathbf{Y}_i$ in $\mathbf{Y}$. We also assume that we know the locations of the segment boundaries $\mathbf{Z}.\ell_1$, $\mathbf{Z}.\ell_2$ and $\mathbf{Z}.\ell$ in the template. Here, $\mathbf{Z}.k$ denotes the k^{th} row in $\mathbf{Z}$. It is a vector of samples corresponding to time k. The prototypical template is initially obtained from an archive.

Procedure : Before applying our method, we remove the baseline wander from the signals. Then, we start the process at some arbitrary point in time p_{start} on the signal that is to be segmented. This need not be an actual segment boundary. We run the algorithm starting at p_{start}, continuously segment $\mathbf{S}$, and add the segments to $\mathbf{Y}$. We also update $\mathbf{Z}$ to reflect the evolution of the time series. This enables us to accommodate gradual changes in the morphology of the signal.

An iteration : We start each iteration with the extraction of a window $\mathbf{W} = \mathbf{S}_{[p_i, p_i+v)}$ from the time series data $\mathbf{S}$ at p_i. Here, the window length is given by $v = \ell + e$, where e is the buffer length. In the following discussion, we use j to index the single parameter signals. The window is then detrended and normalized, and the morphological quality estimates $\{q_j\}_m$ are computed, where q_j represents the morphological similarity between the channel W_j from $\mathbf{W}$ and the corresponding channel from the template Z_j. For each channel j, a pairwise Euclidean distance matrix $\mathbf{pD}^j$ is calculated between Z_j and W_j (Equations 3.1-3.2). The final distance matrix $\mathbf{D}$ is obtained by weighting the pairwise distance matrix $\mathbf{pD}^j$ with q_j (Equation 3.3).

$$c_{x,y} = (W_{x,j} - Z_{y,j})^2 \tag{3.1}$$

$$\mathbf{pD}^j = \begin{bmatrix} c_{1,1} & c_{1,2} & .. & c_{1,|Z_j|} \\ c_{2,1} & c_{2,2} & .. & c_{2,|Z_j|} \\ ... & .. & .. & .. \\ c_{|W_j|,1} & .. & .. & c_{|W_j|,|Z_j|} \end{bmatrix} \tag{3.2}$$

$$\mathbf{D} = \sqrt{\sum_{j=1}^{m} q_j \mathbf{pD}^j} \tag{3.3}$$

The accumulated distance matrix **aD** is then computed from **D** using dynamic programming. We trim **W** to obtain the prefix **W**$*$ so that it contains only the portion of the signal that matches the template. If the prefix **W**$*$ does not contain two segments, the buffer length e is increased and the process is repeated.

We next use the accumulated distance matrix **aD** to find the optimal path alignment between **Z** and **W**$*$, as in DTW. From the alignment, we extract the point $\mathbf{W}.f*$ in **W**$*$ that is matched with the segment boundary $\mathbf{Z}.\ell_2$ in **Z**. This corresponds to the segment boundary we are interested in, because $\mathbf{Z}.\ell_2$ marks the end of the first segment in the template. Then, using the corresponding length $f*$, we update p_{i+1} to $p_i + f*$. Finally $\mathbf{S}_{[p_i,p_{i+1})}$ is added to **Y**. Following the template update, the process is repeated to find the next segment boundary.

3.3.2 weighted time warping

DTW matches two one-dimensional sequences. We extend it to higher dimensions by introducing a weighted norm (Equation 3.3) over the parameter signals to vary the influence exerted by each parameter. The morphological quality metric q_j captures the morphological similarity between the parameter signal W_j and the template Z_j. We discuss the quality estimation in detail in Section 4.

3.3.3 Path Constraint

In a typical formulation of DTW, the distance function that is used to solve DTW (Equation 2.12), allows any path to be taken from (A_1, B_1) to (A_n, B_m). This makes DTW susceptible to degenerate matchings, especially in the presence of noise. For example, a long subsequence of a signal might be matched with a significantly shorter subsequence of another signal.

Therefore, we use local continuity and global path constraints (proposed as Type III and Type IV local continuity constraints in [43]) to prevent such physiologically

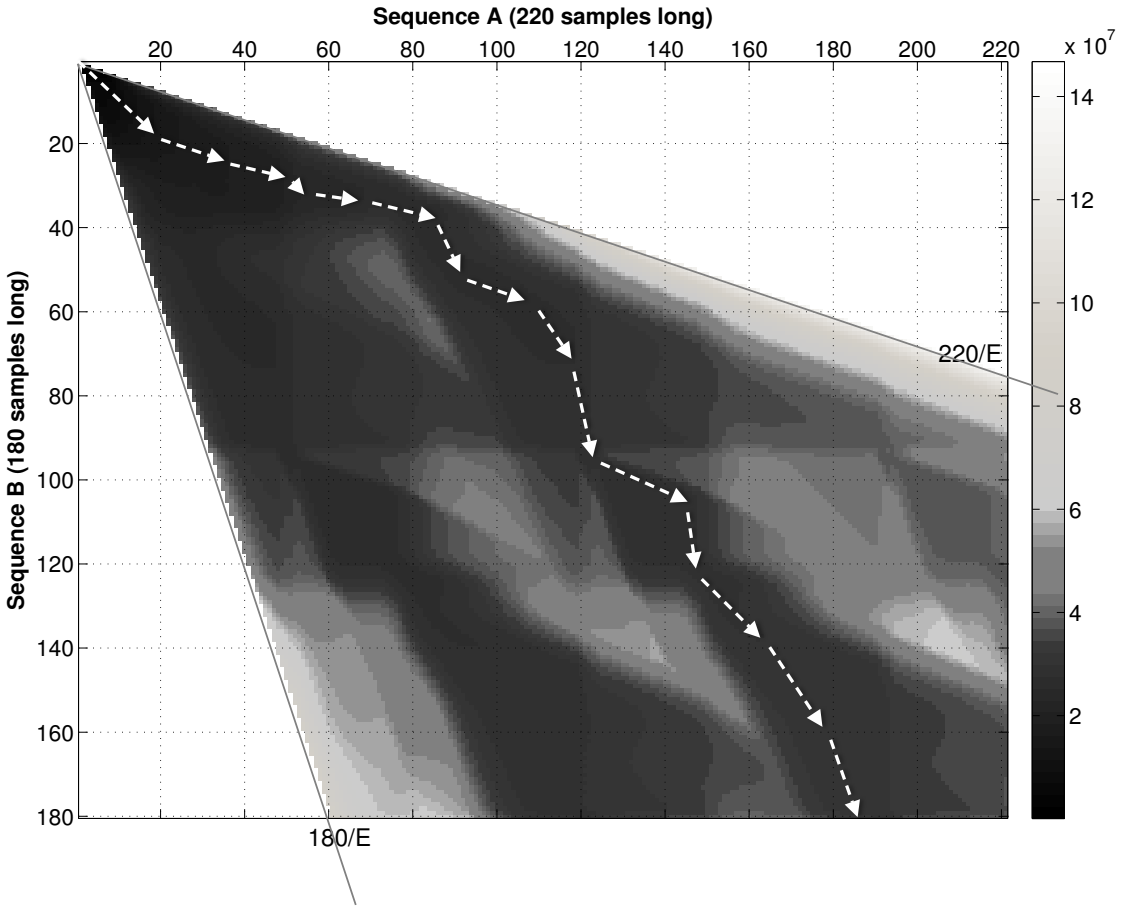


Figure 3-2: The accumulated distance matrix between two sequences A and B when Equation 3.4 is used to compute the accumulated distances. The lighter the region, the higher the cost. Because of the infinitely large cost involved, the optimal path that is shown by arrows, will avoid the white regions. Thereby, the path is constrained globally between the two marked boundaries. Here, sequence A is 180 samples long and sequence B is 220 samples long. Under these constraints, only a subsequence of B that is longer than $180/E$ can legally match the full sequence of A. Similarly only a subsequence of A that is longer than $220/E$ can legally match the full sequence of B. The expansion factor E is 3 for Equation 3.4.

implausible alignments by updating Equation 2.12 with

$$\begin{aligned}\mathbf{aD}(A_i, B_i) = \mathbf{D}(A_i, B_i) + min\{&\mathbf{D}(A_{i-1}, B_{i-1}),\\ &\mathbf{aD}(A_{i-1}, B_{i-2}) + \mathbf{D}(A_i, B_{i-1}),\\ &\mathbf{aD}(A_{i-1}, B_{i-3}) + \mathbf{D}(A_i, B_{i-1}) + \mathbf{D}(A_i, B_{i-2}),\\ &\mathbf{aD}(A_{i-2}, B_{i-1}) + \mathbf{D}(A_{i-1}, B_i),\\ &\mathbf{aD}(A_{i-3}, B_{i-1}) + \mathbf{D}(A_{i-1}, B_i) + \mathbf{D}(A_{i-2}, B_i)\}. \end{aligned} \tag{3.4}$$

This ensures that there are no long horizontal or vertical paths in the matrix **aD** along the alignment. It also results in global path constraints, as shown in Figure 3-2, by excluding certain parts on the accumulated distance matrix in which optimal warping paths could lie.

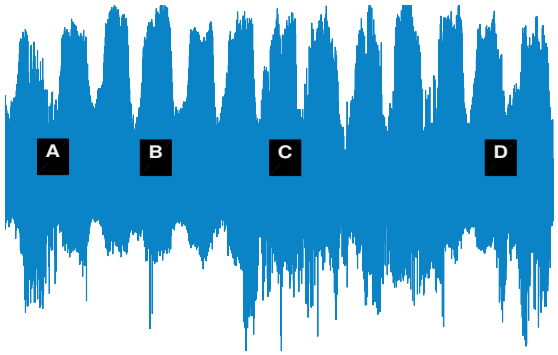
The shortest possible match allowed in Equation 3.4, is a third of the smallest length of the sequence. The typical range of the human heart rate can be bounded by three times of the average heart rate, and one third of the average heart rate. Therefore, while avoiding physiologically implausible matching, it allows us to capture the full valid range of human heart rate. For people with certain cardiovascular conditions, a sudden long pause can occur; our method does not address this special case.

3.3.4 Templates

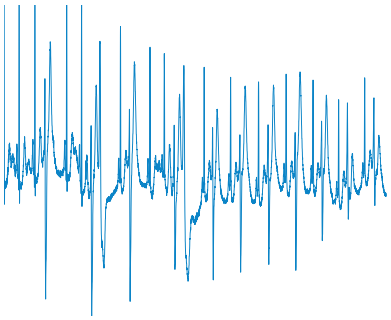
The templates (Figure 3-1) are initially derived from an archive of the prototypical multi-parameter signal. hey are then updated (Figure 3-3) using the recent segment boundaries.

Template Length

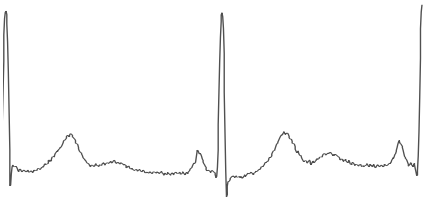
When searching for the segment boundaries we only assume the approximate location of the starting point (p_i). This allows us to start the algorithm at an arbitrary location (p_{start}). It also makes the detection of the segment boundary p_{i+1} less sensitive to



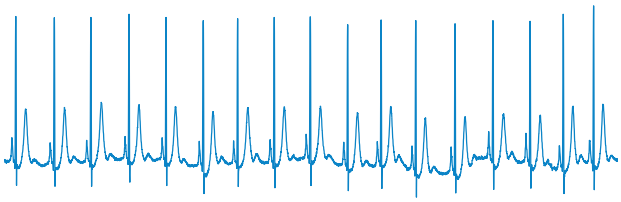
(a) Record 106 from MIT-BIH Arrhythmia database



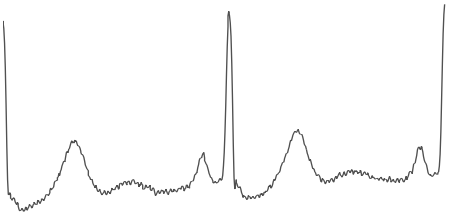
(b) Region C



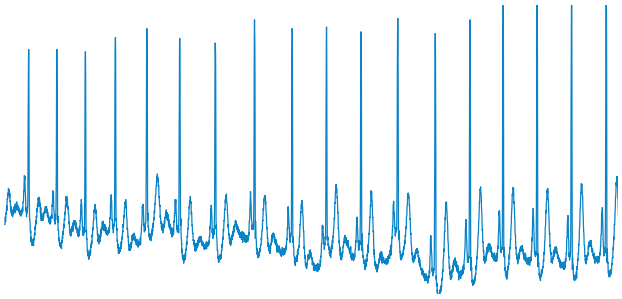
(c) Template over region A



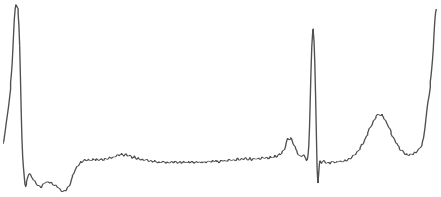
(d) Region A



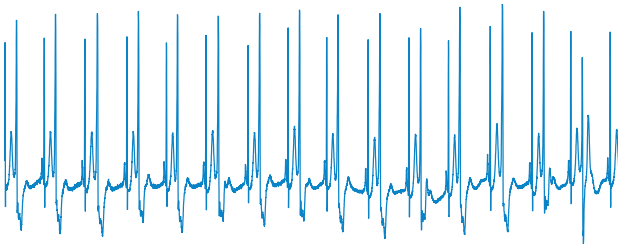
(e) Template over region B



(f) Region B



(g) Template over region D



(h) Region D

Figure 3-3: The evolution of the template over different regions in the record 106 (a) from MIT-BIH Arrhythmia database. In Region A (d), the template (c) reflects the morphology of the signal in that region. When the process moves to Region B (f), the template (e) evolves to follow the morphology in the region. Particularly, S wave in the QRS complex becomes less noticeable. The effect of the template update is prominent in Region D (h). Further, in Region C (f), the template was not updated, because the signal quality was deemed to be poor.

an inaccuracy in determining the previous segment boundary (p_i). To accommodate this we use templates that are more than two segments long.

Template Update

To follow the gradual changes that are common in the physiological signals, we update the template regularly. However, we update the templates only when the variation of the segment lengths in the neighborhood is small, and the quality estimates of all the channels are above a threshold. We require the difference between the maximum segment length and the minimum segment length to be less than 25% of the mean segment length over a moving time period of 60 seconds. We also require the minimum signal quality of any channel in the segment to be greater than a threshold. Empirically we set this threshold at 0.6.

We update the template by averaging the excerpt of the last two segments with the current template time-warped with the excerpt (Equation 3.6). The time warping is necessary, since the current template $\mathbf{Z}*$ and the excerpt of the last two segments $\mathbf{Z}'$ can be of different lengths. Finally, we normalize the template by the sum of the absolute value of the samples.

$$\mathbf{Z}' = \mathbf{S}_{[(p_{(i-1)}-\epsilon),p_{(i+1)}]}; \epsilon \geq 1 \tag{3.5}$$

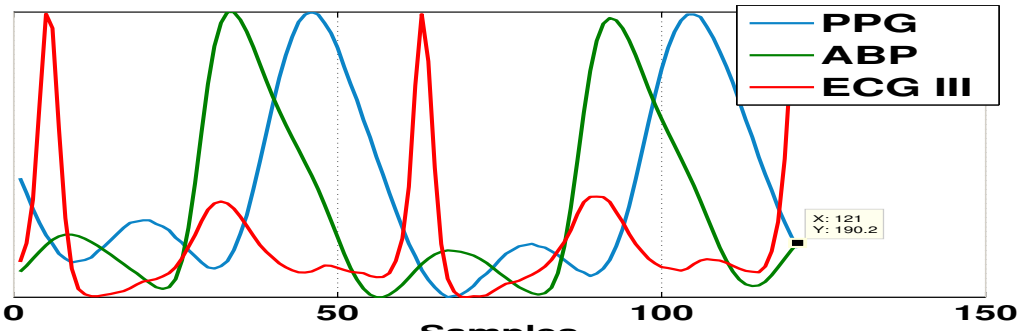
$$\mathbf{Z}'' = warp(\mathbf{Z}) + \eta\mathbf{Z}'; \eta \geq 0 \tag{3.6}$$

$$\mathbf{Z}_j* = \frac{\mathbf{Z}''_j}{\sum_i |Z_{(j,i)}|} \forall j = 1, 2..m \tag{3.7}$$

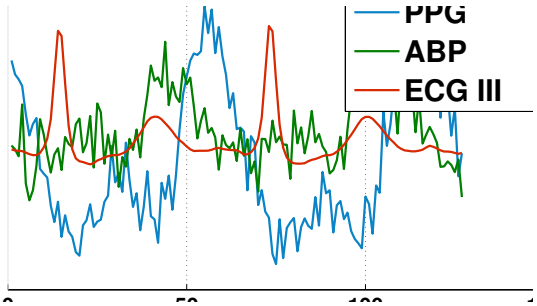
Here, ϵ ensures that the template consists of at least two segments. We vary the influence of the recent segments on the template through the constant $0 \leq \eta \leq 1$, where $\eta = 0$ implies a static template. We chose $\eta = 0.06$, but the results are not sensitive on the choice of η.

3.3.5 Long Segments

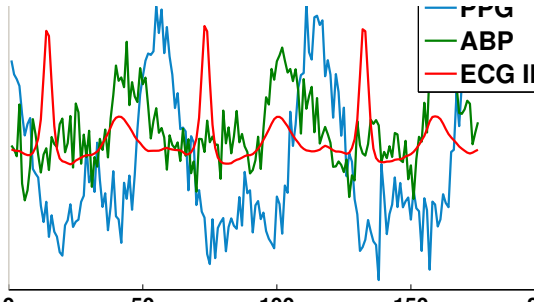
Spurious matching occurs when the segments in the window $\mathbf{W}$ are significantly longer than the segments in the template $\mathbf{Z}$; therefore, the current buffer length e is not long



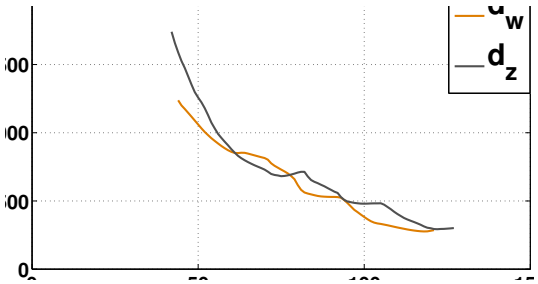
(a) Template $\mathbf{Z}$ of 121 samples long



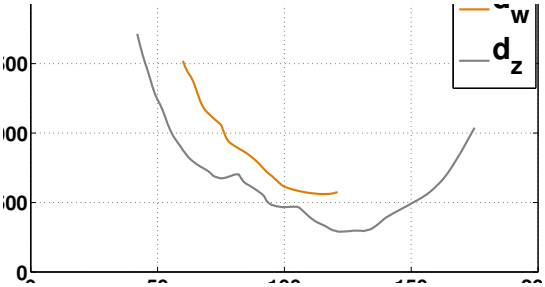
(b) Window $\mathbf{W}_1$ of 126 samples long



(c) Window $\mathbf{W}_2$ of 174 samples long



(d) Cost of matching for $\mathbf{W}_1$



(e) Cost of matching for $\mathbf{W}_2$

Figure 3-4: The template $\mathbf{Z}$ (a) is initially matched with a window $\mathbf{W}_1$ (b). However, the minimum cost to match the full template to a prefix of the window (min d_z) is greater than the minimum cost to match the full window to the prefix of the template (min d_w) (d). This is considered a spurious matching. Therefore, the process is repeated with a longer window $\mathbf{W}_2$ (c). For $\mathbf{W}_2$, the minimum cost to match the full template to a prefix of the window (min d_z) is less than the minimum cost to match the full window to the prefix of the template (min d_w) (e). Thus, the desired matching is obtained.

enough to span at least two segments in the window. In such cases, the window $\mathbf{W} = \mathbf{S}_{[p_i, p_i+v)}$ would not contain two full segments and does not give the expected matching. In such situations, we have to repeat the matching with a longer window (Figure 3-4).

We use the following equations on the accumulated distance matrix **aD** to check for spurious matching.

$$d_z* = \min_{k_1} \frac{1}{k_1} \mathbf{aD}(|\mathbf{Z}|, k_1) \tag{3.8}$$

$$z* = \operatorname*{argmin}_{k_1} \frac{1}{k_1} \mathbf{aD}(|\mathbf{Z}|, k_1) \tag{3.9}$$

$$d_w* = \min_{k_2} \frac{1}{k_2} \mathbf{aD}(k_2, |\mathbf{W}|) \tag{3.10}$$

$$w* = \operatorname*{argmin}_{k_2} \frac{1}{k_2} \mathbf{aD}(k_2, |\mathbf{W}|) \tag{3.11}$$

Here d_z* is the minimum cost to match the full template ($\mathbf{Z}$) with any prefix string ($\mathbf{W}_{[0,w*]}$) of window $\mathbf{W}$. Similarly, d_w* is the minimum cost to match the full window with any prefix string of the template. Further, k_1 and k_2 are the proxies for the corresponding alignment lengths, $|\mathbf{Z}|$ is the length of the template and $|\mathbf{W}|$ is the length of the window.

If $d_z* < d_w*$ then the window is shorter than two segments. Therefore, we increase v through e and repeat the process with a longer window. Otherwise, we trim the window and obtain $\mathbf{W}*$, the portion of the window that matches the template.

3.4 Experimental Results

In this section, we present the experimental results for our method. We begin with the description of our dataset and then present the performance of the segmentation method.

We applied our method to multi-parameter physiological signal data (Figure 3-5) from MIMIC at Physionet.org [21]. The database has 72 waveform records with several annotation sets including ECG beat labels. It includes recordings from multiple

ECG channels, Arterial Blood Pressure (ABP) and Photo Plethysmogram (PPG). The signals are sampled at 125 Hz. From this database, we selected all 70 records that contained one continuous hour of at least one ECG channel, and ABP, PPG or both signals.

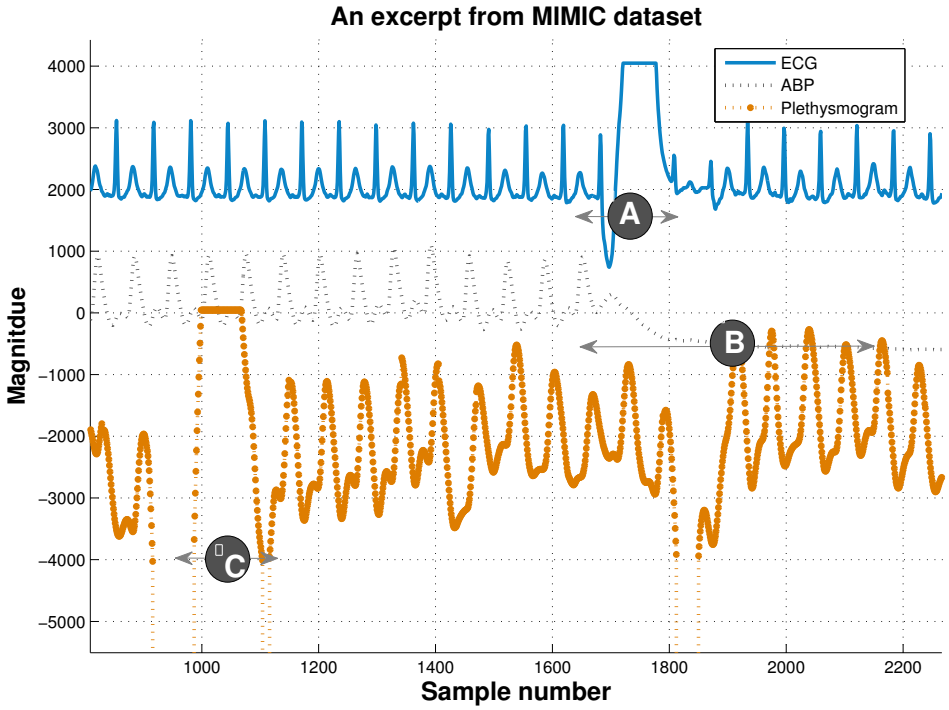


Figure 3-5: The Electrocardiogram (ECG), Arterial Blood Pressure (ABP) and the Photo Plethysmogram (PPG) extracted from MIMIC, record 039. Over time period A, ECG is corrupted. Over time period B, ABP is unavailable. Over time period C, PPG is corrupted.

We compared our method to a widely used QRS detection based segmentation method [45, 55]. This method provides high specificity and sensitivity with low computational load [34]. Further, it is publicly available in the open source software package EPlimited[2].

We conducted the following experiments

1. Raw Data : on the raw MIMIC data we compared our algorithm to the QRS detector (test 1 and 2).

[2] `http://www.eplimited.com/`

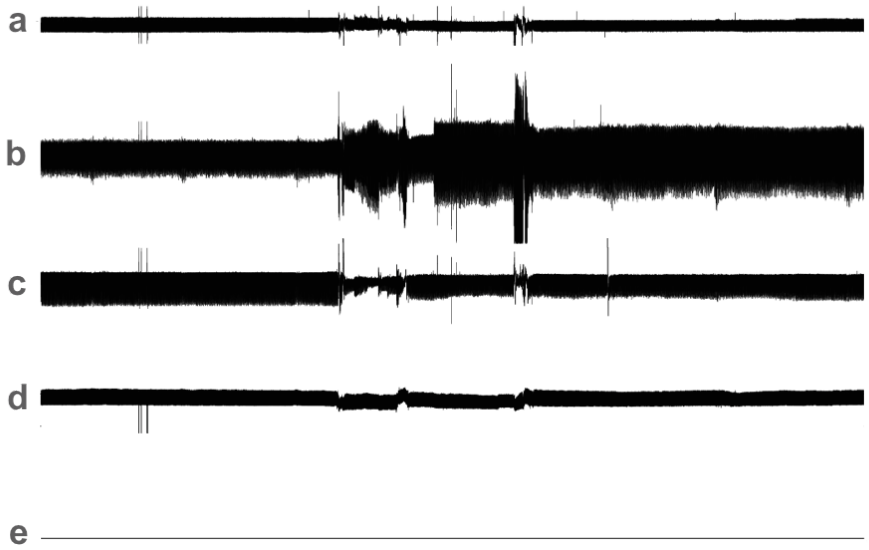


Figure 3-6: Record 213 contains a) ECG channel I, b) ECG channel II, c) ECG channel V, d) ABP and e) PPG. The plethysmogram is completely absent over the 12 hour period. A severe corruption spans across all the channels for significant amount of time also.

2. Synthetic Data

- AWGN noise : we selected clean excerpts from the raw data, and artificially corrupted them with AWGN noise (test 3 and 4).

- Transient Corruption : we selected clean excerpts from the raw data, and artificially corrupted them with transient corruption (test 5).

We evaluate the performance using the number of errors and the average error in milliseconds. The number of errors is defined as the total number of the estimated segment boundaries that are different from the actual boundaries by more than one sample. This also includes the cases where a segment boundary is completely missed and an erroneous boundary is found. The average error is the mean distance between an actual segment boundary and the closest segment boundary found by the method.

3.4.1 Experiment 1 : Raw MIMIC Data

We performed two tests on the raw data. In the first test (test1), we randomly selected 10 records and ran the tests for the minimum of 12 hours and the end of the record. We compared the results of our method and the QRS detector to the segment boundaries marked in the data.

The results are presented in Table.3.2. The results exhibit significant inter-record differences. This is mainly due to the presence of sporadic corruptions in some records, which results in a burst of errors in these records. For example, in record 213 (Figure 3-6), the corruption is severe, and it spans all the channels. It causes significantly large number of errors for both our method and the QRS detector. The median number of errors for our algorithm was 65, whereas it was 330 for the QRS detector.

Table 3.2: Test 1: Comparison of WTW against the QRS detector on 10 randomly selected records from the raw MIMIC data.

		WTW (ours)		**QRS**	
Rec ID	avg Sgmt len (ms)	Err #	avg Err (ms)	Err #	avg Err (ms)
39	506.25	6	0.01	30	0.25
41	737.91	2	0.00	31	0.47
55	671.13	19	0.05	385	28.27
208	642.67	215	20.63	699	135.39
216	771.33	293	1.10	330	1.64
220	632.88	5	1.65	4	1.20
253	864.40	35	0.15	346	35.13
262	824.93	65	0.30	249	3.55
284	651.70	119	0.23	191	0.96
226	532.17	311	0.55	675	6.25
213	660.31	1921	3.19	2927	42.22
Median number of beats in one record is 56007					

In the second test (test 2), we ran the test across all 70 records for an hour. The results are presented in Figure 3-7. WTW was able to segment 4.121×10^5 segments accurately while the QRS detector was able to segment 4.111×10^5 segments correctly. Although our method made fewer mistakes compared to that made by the

QRS detector, the difference in the accuracies of both the methods is not significant because of the sparsity of the occurrence of corruption in the raw data.

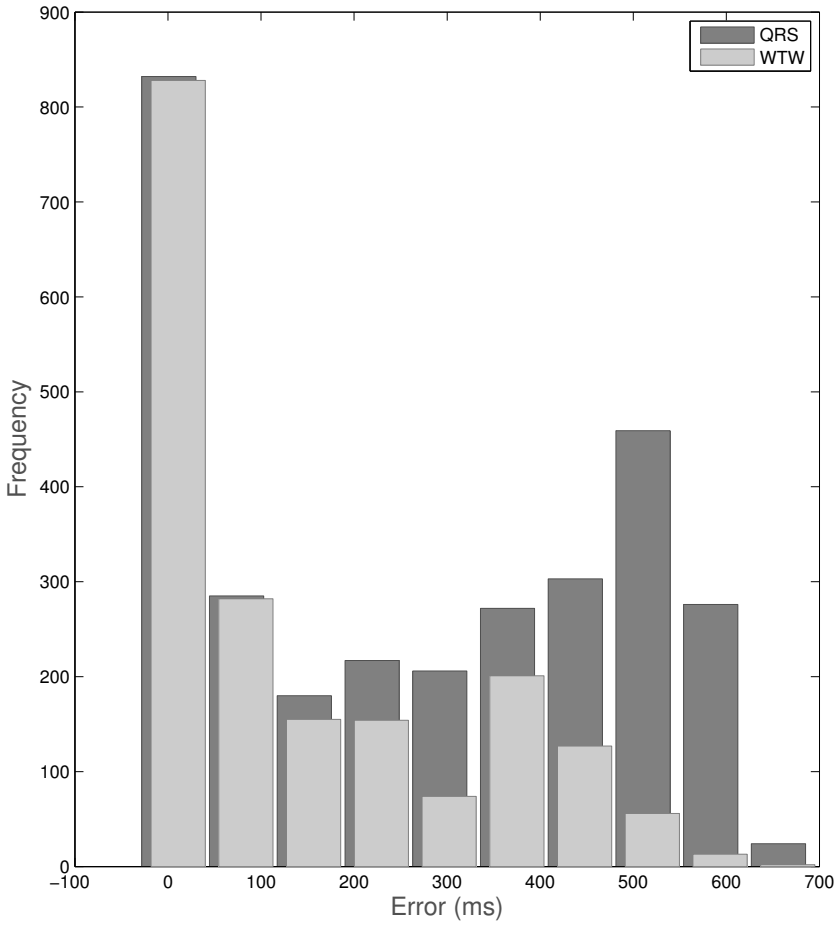
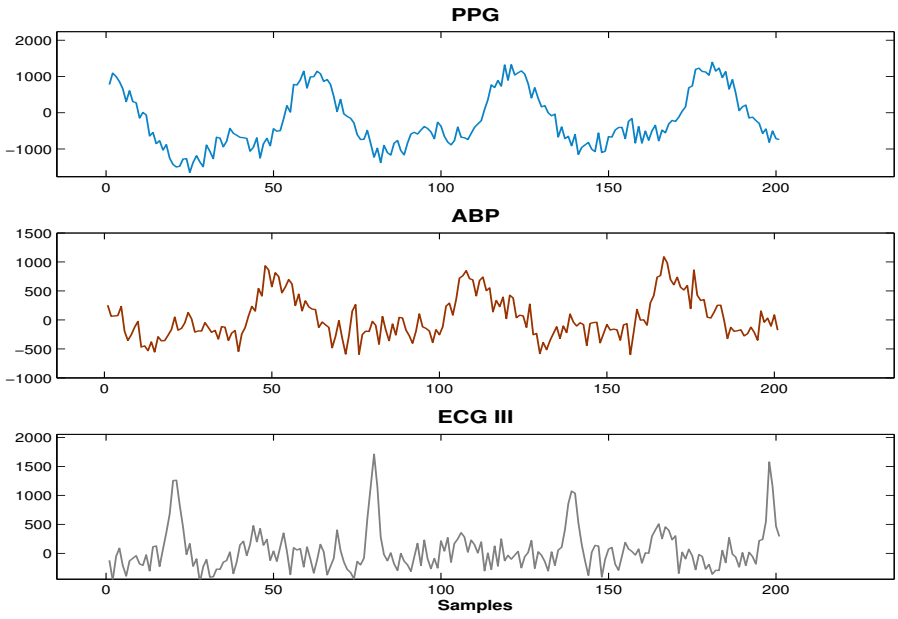


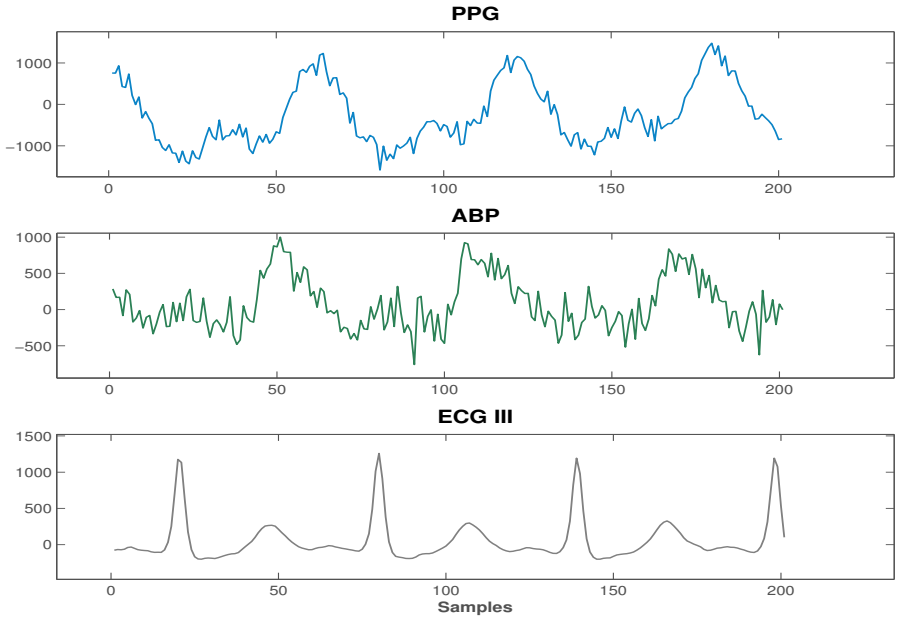
Figure 3-7: Test 2: distribution of errors on the raw data. WTW was able to segment 4.121×10^5 segments accurately while, the QRS detector was able to segment 4.111×10^5 segments correctly. It should be noted that, while the QRS detector is using only one ECG channel, WTW is making use of other channels as well.

3.4.2 Experiment 2: Artificially Corrupted MIMIC Data - Additive Noise

In this experiment, we evaluated our algorithm's performance on noisy data. We obtained clean excerpts from the raw MIMIC data and corrupted it with artificial



(a) AWGN on all 3 parameter signals



(b) AWGN on all but the last parameter signal

Figure 3-8: (a) Test 3 is carried out on the data altered with AWGN on all m parameters. (a) In Test 4 only $m-1$ parameters are altered with AWGN.

Table 3.3: Test 3, 4, and 5: Summary of the experimental results on the artificially corrupted data.

Average error (ms) on artificially corrupted data				
Type		**QRS**	**WTW**	
			all m	any $m-1$
AWGN	20 dB	12.87	0.87	0.0097
(SNR)	10 dB	188.11	3.27	0.0011
	0 dB	303.48	5.81	0.008
Transient Corruption		387.32	-	2.89
Average segment length is 521 ms.				

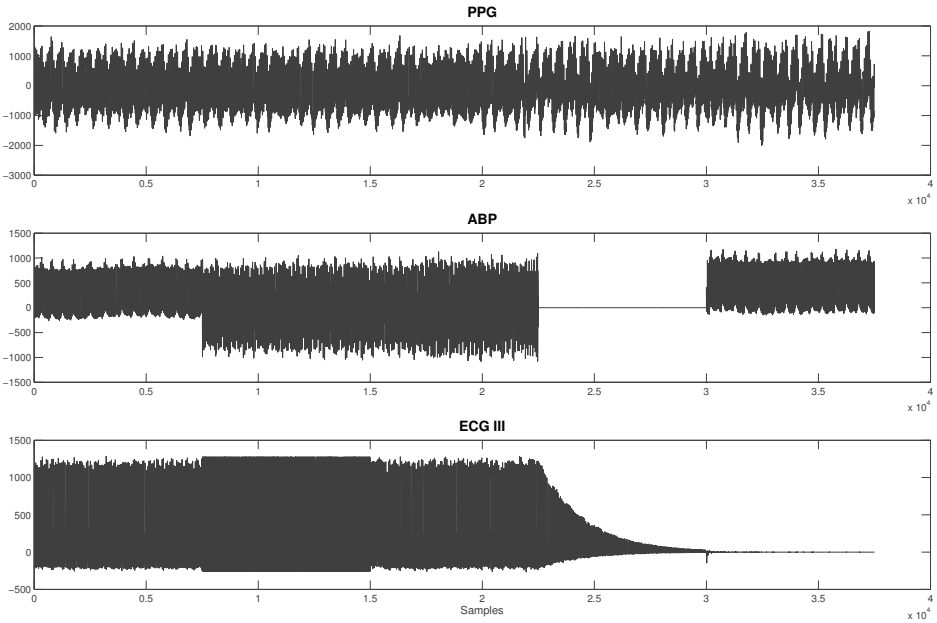
noise. We randomly extracted 1000 high quality 5-minute excerpts for which both our method and the QRS detector were 100% accurate. We added AWGN noise at different Signal/Noise (SNR) levels to these excerpts and tested both the methods on them.

We calculate the power of the noise added $P_n = P_s - SNR$, using the signal variance $P_s = 20log(std(S))$ instead of the absolute signal power $P_s = 10log(\frac{\sum_{i=1}^{n} S_i^2}{n})$ to remove the effects of the offset.

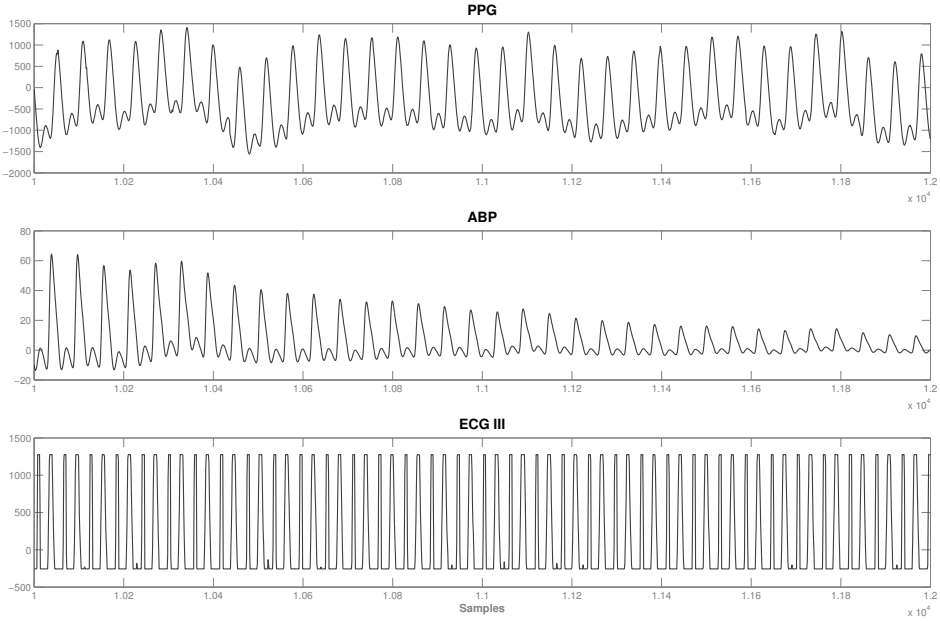
We carried out two tests. In the first test (test 3), we added noise to all m channels, whereas in the second test (test 4), we added noise to only $m-1$ channels. The results of these tests are presented in Table 3.3. Our method was able to identify the segment boundaries with reasonable accuracy even in the presence of significant additive noise; further, if any one of the channels is free of noise, the performance is comparable to the performance on the clean raw data.

3.4.3 Experiment 3: Artificially Corrupted MIMIC Data - Transient Corruption

In this test (test 5) we evaluated our algorithm's performance on the data altered with transient corruption. Again, we randomly extracted 1000 high quality 5-minute excerpts and artificially corrupted them. On each of these excerpts, we corrupt $m-1$ single parameter signals. For each these single parameter signals, we randomly se-



(a) Synthetic transient corruption on a 5 minute long excerpt



(b) Transient corruption over a 16s long region

Figure 3-9: An example distribution of the synthetic corruptions applied on an excerpt. A high quality 5 minute long excerpt from Record 039 is altered with transient corruption. (a) There are $m = 2$ single parameter signal. Two of them are corrupted with signal interruption, exponential damping, overshooting and clipping, superimposition of high frequency sinusoidal signal, and superimposition of low frequency sinusoidal signal. In this particular example, PPG signal was not corrupted. (b) The effect of corruption is clearly noticeable in this 16s long region. ABP is corrupted with exponential damping. ECG III is corrupted with amplification and clipping. PPG is left out of the corruption.

lected 5 non-overlapping 1 minute long regions and randomly applied one of the following types of corruptions: signal interruption, exponential damping, overshooting and clipping, and superimposition of artificial low frequency signals and high frequency signals. The ECG channels and one of ABP or PPG were corrupted.

The results are presented in Table 3.3. Under transient corruption, the average error was 2.89 ms for WTW. This is comparable to that of the data with AWGN at SNR 10 dB on all m parameters. Under severe transient corruption, the QRS detector becomes totally unusable.

3.5 Summary

In this chapter, we presented a novel online method for temporal segmentation of quasiperiodic multi-parameter physiological signals in the presence of noise and transient corruption. Our method uses weighted time warping to exploit the relationship between the partially correlated channels and the repetitive morphology of the time series. Our method has a greater constant overhead in computational complexity relative to QRS detection based segmentation algorithm. For a window of length ℓ, our method uses dynamic time warping which runs in $O(\ell^2)$ time and space. A QRS detection based segmentation runs in $O(\ell)$ time. In the examples used in this paper ℓ is less than 500.

Our method is particularly useful when the system suffers noise and transient corruption. For corruptions, we tested our method on few artificial corruptions; but the real world corruptions could be different, and perhaps adversarial. We haven't tested our method on these specific types of corruptions. In the case of additive noise, AWGN is the most difficult to handle because it spans the entire frequency spectrum and has the highest uncertainty. Our method performs well against AWGN.

We chose ABP, PPG and ECG for testing, because they are commonly available in the recordings of ICU patients. However, our method should be applicable to any set of correlated physiological signals such as Central Venous Pressure (CVP), Pulmonary Arterial Pressure (PAP) and respiratory signal.

We showed that our method performs as well as an excellent QRS detector on relatively clean ECG data. On 1 hour long test data across 70 records, our method achieved 99.56% accuracy, whereas the QRS detector achieved 99.41% accuracy. When AWGN is synthetically added, the difference in the performance between our method and the QRS detector becomes significant. Our method was able to limit the average error to 5.81 ms when all m parameters were corrupted with AWGN at SNR: 0dB, and to 0.008 ms when $m - 1$ parameters were corrupted at the same SNR. The average error for the QRS detector rose to 303.48 ms when the ECG channel is corrupted with the same AWGN. Similar results were observed on transient corruption.

Chapter 4

Quality Estimation

This chapter discusses the assessment of signal quality in our framework. After giving an introduction, we relate to the background. In Section 4.3, we present a novel signal quality metric that can be used across different types of signals. Finally, in Section 4.4, we explain how our system uses the signal quality estimates.

4.1 Introduction

Our framework requires the assessment of the signal quality at multiple stages of the system. In the joint segmentation of the multi-parameter signals (Chapter 3), we weigh the individual signal's influence on the joint segmentation of the multi-parameter signal by the estimate of the quality of the signal. In the reconstruction of the morphological features of the physiological signals (Chapter 5), we use the signal quality estimates to identify the presence of noise and corruption in the signal. In Chapter 6, we use the signal quality of the reconstructed signals to combine the reconstructions from multiple sources.

If we had the model of the source, the signal quality assessment would be straightforward, the signal to noise ratio (SNR) gives an estimate of the signal quality. SNR is the ratio between the source signal power, and the difference between the observed signal power and source signal power. However, for physiological signals, the source signal power is hard to obtain. Therefore, researchers use indirect measures to esti-

mate the signal quality of the physiological signals. They use various characteristics of the physiological signals to identify the presence of the artifacts and the level of corruption. Here, the assumptions are that the characteristics of the signals are known *a priori*, and the noise artifacts cause any deviation present in the signal.

The signal quality estimates are calculated for a region in the signal, ranging from a single sample to tens of seconds long windows. Our framework operates at the granularity of a heartbeat. Hence, we require the signal quality estimates at heartbeat granularity. In addition, we have the following requirements.

- We require the signal quality estimates to be independent of the beat type and the location of the segment. For instance, a segment with significant noise must have a lower signal quality than a clean segment irrespective of the location, signal amplitude, segment length, and the type of the beat.

- We require the signal quality estimates to be comparable across different type of signals, such as ECG, ABP and PPG signals. For a multi-parameter signal, the signal quality estimate is computed separately for each signal. These signal quality estimates must be comparable, i.e., using the estimates, we should be able to assess the individual signal's relative quality.

4.2 Previous work

Researchers have come up with various the signal quality estimates for various physiological signals including ECG, ABP and PPG [11, 38, 44, 63]. ECGSQI estimates the signal quality of a multi-channel ECG signal [38]. ABPSQI [38] combines wSQI and jSQI to obtain the arterial blood pressure signal quality. PPG signal quality [13] estimation is done using Hjorth [31] parameters. These methods generate independent quality estimations, and hence they are not comparable with each other.

4.3 New morphological dissimilarity Metric

We hypothesize that the corrupted regions of the signals will be morphologically dissimilar from clean signals. We use an evolving template to represent the hypothesized corruption-free signal (Section 3.3.4). The templates (Figure 3-1) are initially derived from an archive of the prototypical multi-parameter signal. They are then updated using the recent signal estimates. We then compute the morphological dissimilarity between a window from the signal, and the template. The inverse of the morphological dissimilarity gives the quality estimate. Ideally, we expect the signal quality estimates derived from morphological dissimilarity to provide the ordering in Figure 4-1.

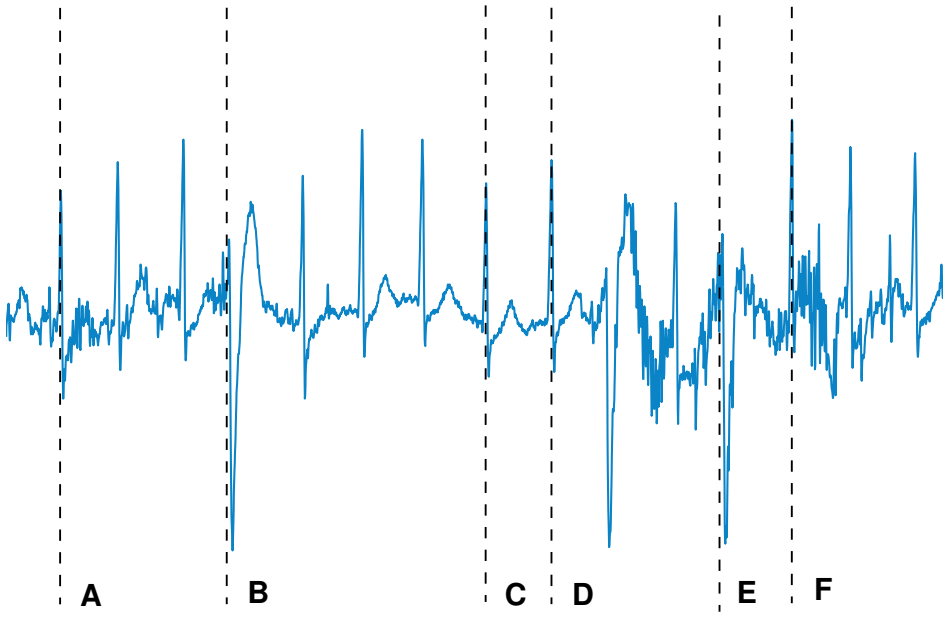


Figure 4-1: Record 200 from MIT-BIH Arrhythmia Database. Ideally, we expect signal qualities at segments $A - F$ to be $[C, D] > B > [A, F] > E$. The rationale is that two normal beats will be more similar to each other compared to a PVC beat. A normal beat with noise must be more dissimilar than a beat with PVC and a normal beat. Finally, when there is a corruption or signal interruption, the signal quality must be the lowest.

4.3.1 Method

Consider a multi-parameter signal **S** and the template **Z**. The signal quality of the k^{th} signal at segment p_i is calculated from the morphological dissimilarity between the window $A = S^k_{[p_i,p_{i+1})}$, and the corresponding signal in the template $B = Z^k$. Here A and B are one-dimensional signals of length n and m respectively.

$$c_{i,j} = (A_i - B_j)^2 \tag{4.1}$$

$$\mathbf{D} = \begin{bmatrix} c_{1,1} & c_{1,2} & .. & c_{1,m} \\ c_{2,1} & c_{2,2} & .. & c_{2,m} \\ ... & .. & .. & .. \\ c_{n,1} & .. & .. & c_{m,n} \end{bmatrix} \tag{4.2}$$

We obtain the morphological dissimilarity by computing the warped distance between two sequences, for example, the window from the signal and the template. Using dynamic programming, we compute the accumulated distance matrix **aD** (Equation 4.3). The morphological dissimilarity is given by $md = \mathbf{aD}(A_n, B_m)$.

$$\begin{aligned} \mathbf{aD}(A_i, B_i) = \mathbf{D}(A_i, B_i) + min\{&\mathbf{D}(A_{i-1}, B_{i-1}), \\ &\mathbf{aD}(A_{i-1}, B_{i-2}) + \mathbf{D}(A_i, B_{i-1}), \\ &\mathbf{aD}(A_{i-1}, B_{i-3}) + \mathbf{D}(A_i, B_{i-1}) + \mathbf{D}(A_i, B_{i-2}), \\ &\mathbf{aD}(A_{i-2}, B_{i-1}) + \mathbf{D}(A_{i-1}, B_i), \\ &\mathbf{aD}(A_{i-3}, B_{i-1}) + \mathbf{D}(A_{i-1}, B_i) + \mathbf{D}(A_{i-2}, B_i)\}. \end{aligned} \tag{4.3}$$

The inverse of md gives the signal quality estimate q (Equation 4.4).

$$q = \frac{1}{md} \tag{4.4}$$

We prefer Derivative Dynamic Time Warping (DDTW) to DTW, for computing the warped distance. The reasoning behind this is that if a local feature (e.g. valley) is more prominent in one sequence than in the other sequence, when using DTW,

this would result in extreme warping to minimize the global distances between two sequences. DTW only considers the data-points in Y-axis. In contrast, DDTW considers the first derivative, which include the changes (variability) in the Y-axis as well.

Therefore instead of using the sample values, we use the first derivative of them

$$A'_t = \frac{(A_t - A_{t-1}) + (A_{t+1} - A_{t-1})/2}{2} \qquad 1 < t < n \tag{4.5}$$

$$B'_t = \frac{(B_t - B_{t-1}) + (B_{t+1} - B_{t-1})/2}{2} \qquad 1 < t < m \tag{4.6}$$

$$c_{i,j} = (A'_i - B'_j)^2 \tag{4.7}$$

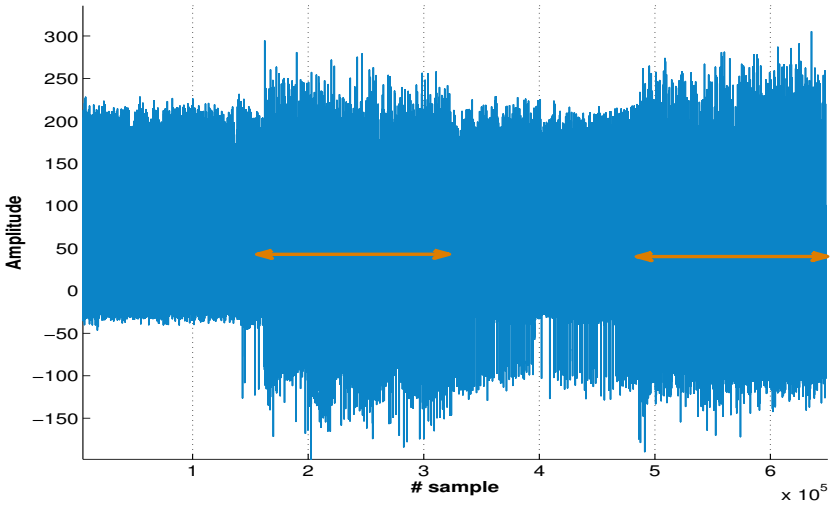
$$\tag{4.8}$$

Here $A \in \Re^n$ and $B \in \Re^m$ are two sequences, and we want to compute the morphological dissimilarity between them; A' and B' are the derivatives of the original sequences.

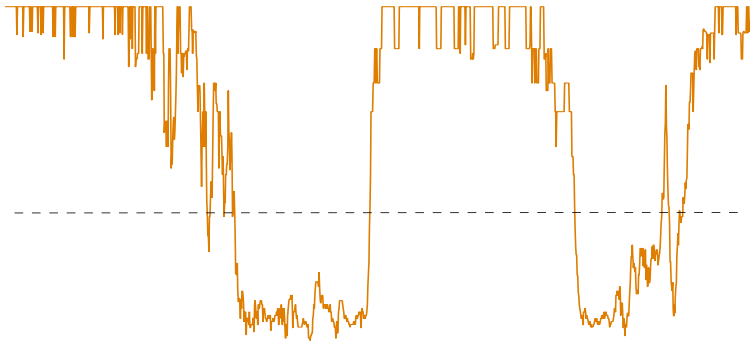
In DDTW computation, we again employ the path constraints (Section 3.3.3) to avoid degenerate matching. Taking the derivative usually amplifies the distortions present in the signal. Therefore, DDTW helps us find the corrupted regions better. However, the amplification of distortions also increases the probability of the degenerative matching. Therefore, the path constraints are essential. The morphological dissimilarity generated by DDTW is unbounded, and is dependent on the scaling of the samples.

4.3.2 Comparison

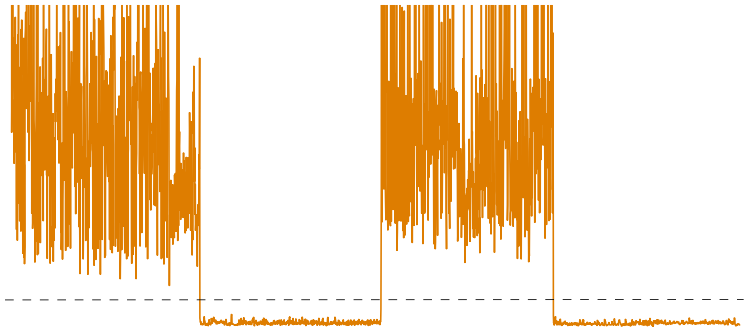
morphological dissimilarity metric estimates the quality of the signal beat by beat. In contrast, previously discussed methods estimate the quality of the signal over a region [38]. For instance, ECGSQI is calculated on a 10 s window. The ability to obtain a more fine-grained quality estimate helps us handle the bordering regions (start and end of the corrupted region) better. For instance in Figure 4-2, the SQI obtained from morphological dissimilarity is able to identify the start and the end of the corrupted



(a) AWGN added signal data



(b) ECGSQI



(c) SQI from morphological dissimilarity

Figure 4-2: Record 201, channel MLII from MIT-BIH Arrhythmia database (a). The original signal is added with AWGN at 0dB between 8 to 15 minute and 23 to 30 minutes. For the corrupted signal, SQI using morphological dissimilarity (c) and ECGSQI (b) are computed.

region better than ECGSQI. Further, when the R-peaks in the ECG signals are clearly identifiable, ECGSQI tends to overestimate the signal quality irrespective of the distortion in the overall morphology of the beat; it is noticeable during last few hundred beats of the record where both *wqrs* and *ep_limited* agreed mostly.

Unlike the related methods, the morphological dissimilarity is applicable to all of, ECG, ABP and PPG signals. Therefore, we are able to use a single metric across different types of signals (Figures 4-3, 4-4). Particularly, in Section 3.3.2, we use the relative signal quality estimates obtained from the morphological dissimilarity to weigh the individual signal's influence on the joint segmentation of a multi-parameter signal. We normalize the signal qualities estimated for individual signals by their sum to obtain the relative signal quality estimates (RSQE). Other previously discussed methods cannot be directly used for this purpose.

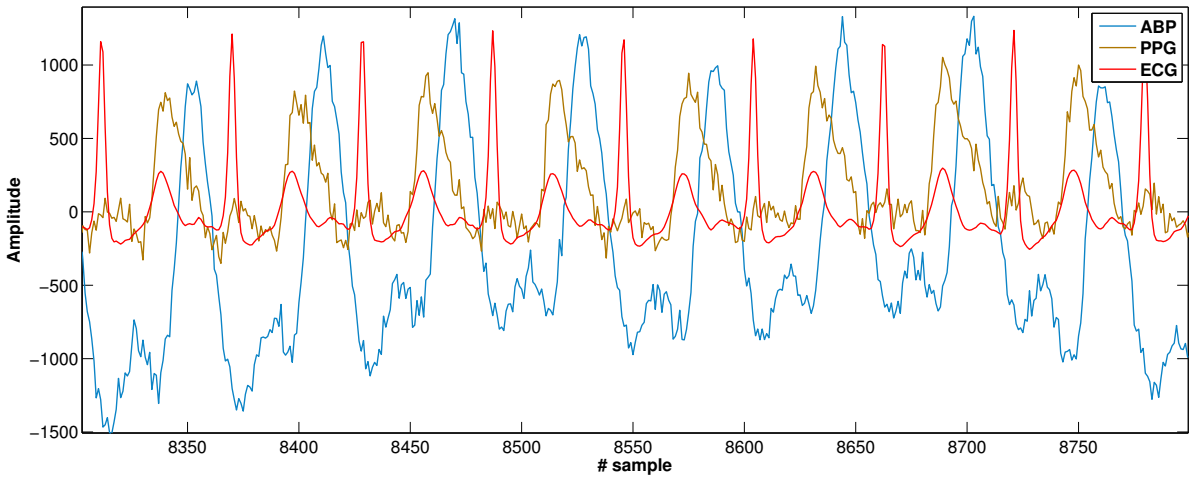
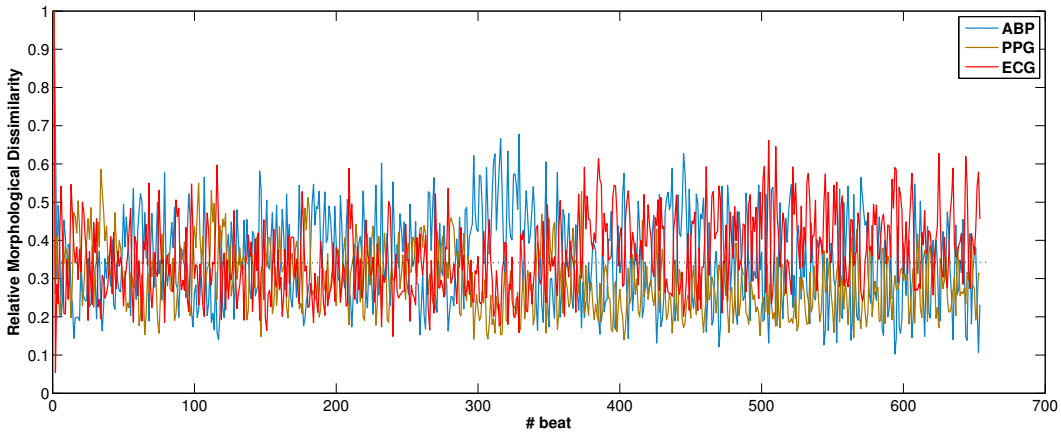


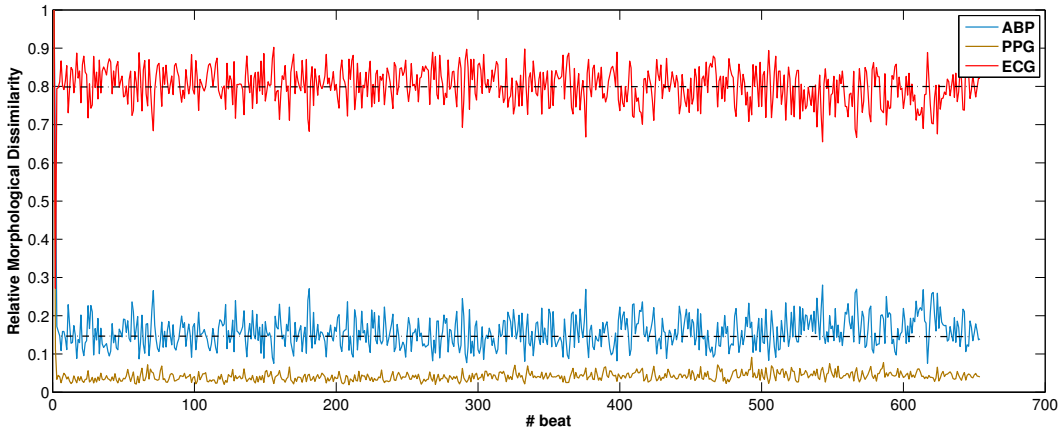
Figure 4-3: An example recording of a multi-parameter physiological signal (record 039 from MIMIC) with 20dB AWGN added to ABP and PPG signal. This record is used illustrate the application of RSQEs in Figure 4-4.

4.3.3 morphological dissimilarity Metric with LCSS

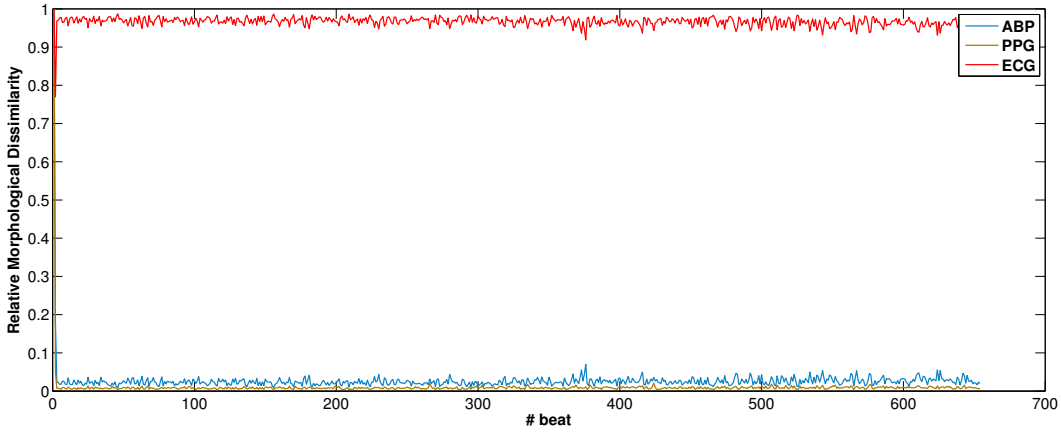
When using the morphological dissimilarity obtained from the DTW distance on the gradient of the signal samples to estimate the quality, we observe that it can accurately estimate the relative signal quality (Figures 4-3, 4-4). However, it is unbounded and



(a) Signals are corruption-free



(b) ABP and PPG signals are added with 20dB AWGN



(c) ABP and PPG signals are added with 10dB AWGN

Figure 4-4: Relative Signal Quality Estimates (RSQE) of a multi-parameter physiological signal (Figure 4-3). When all three signals are corruption-free, RSQEs are approximately equal, and take an average of 0.33. When 20dB AWGN is added to ABP and PPG signals, it results in an RSQE 5 times higher (average RSQE for ECG is 0.8) for ECG than others (average RSQE for ABP is 0.16). At 10dB AWGN the difference is significant.

dependent on the scaling of the sample values. Therefore, we cannot infer a hard threshold to identify the regions of corruptions. For example, in Figure 4-2(c), the morphological dissimilarity has a large variation.

Therefore, we use LCSS, instead of DTW, to estimate the morphological dissimilarity. We still use the gradient of the samples and take the variability in y-axis into account.

$$LCSS(A'_x, B'_y) = \begin{cases} 0 & \text{if A or B is empty} \\ 1 + LCSS(A'_{x-1}, B'_{y-1}) & \text{if } |A'_x - B'_y| < \delta_y \text{ and } |x - y| < \delta_x \\ max\{LCSS(A'_{x-1}, B'_y) & \\ LCSS(A'_x, B'_{y-1})\} & \text{otherwise} \end{cases} \tag{4.9}$$

$$q = \frac{LCSS(A, B)}{min(n, m)} \tag{4.10}$$

$$\tag{4.11}$$

Furthermore, using LCSS has several other advantages. It is bounded, and hence, is comparable across beats at different locations and of different types. It lets us obtain a more accurate signal quality estimate (Figure 4-5) at a finer granularity of heartbeat.

4.3.4 Adaptive Signal Quality Estimates

To estimate the signal quality, we compare a window of the signal against the template. This would be valid if any difference between the template and the window were due to the noise or corruption presents in the signal. However, the signal morphology evolves along the time. For instance, heart rate may increase, the amplitude of the ABP signal might drop, etc. We regularly update the template to accommodate the time evolution of the signal. The template update was discussed in Section 3.3.4.

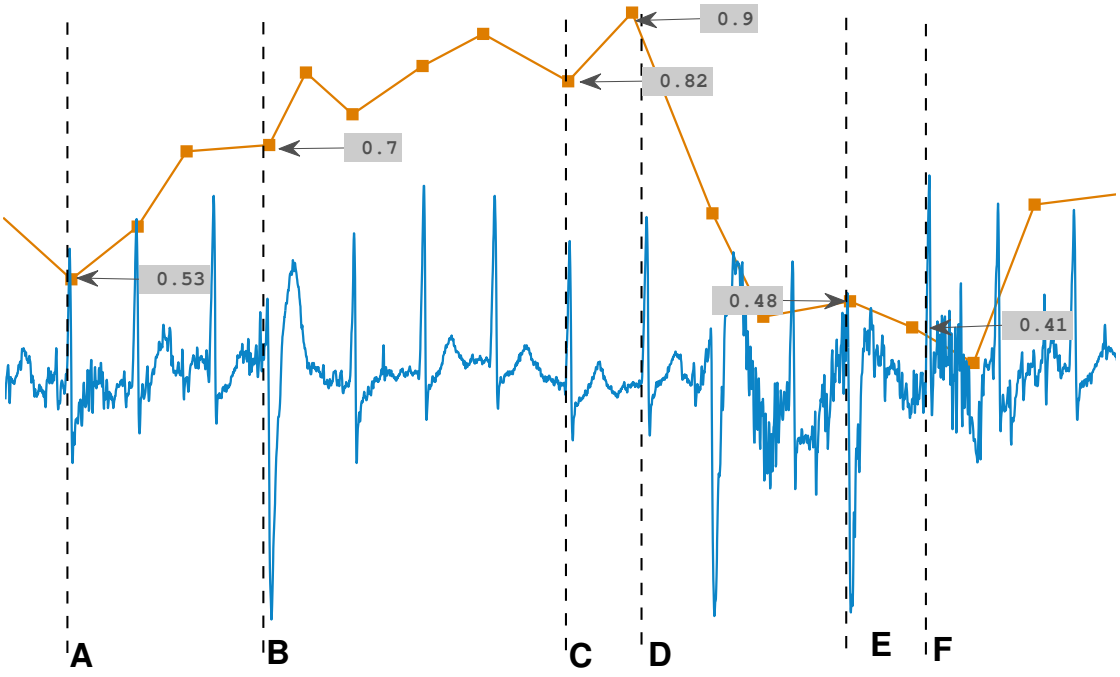


Figure 4-5: For the example in Figure 4-1, the SQE estimated using LCSS. The SQE generated by LCSS is bounded between 0 and 1. Through LCSS we achieve the signal quality estimates in the order $[C, D] > B > [A, E] > F$, which is close to the ideal situation.

4.4 Implementation

First, in the joint segmentation of the multi-parameter physiological signals (Section 3.3.2), we require the Relative Signal Quality Estimates (RSQE) beat by beat. We compute the morphological dissimilarity between the window and the current template, and obtain the signal quality estimate (Equation 4.10). We normalize the signal quality indices across all the signals by their sum. The resulting relative signal quality estimates $\{q_j\}_m$, are then used in WTW (Equation 3.3).

In Chapter 5, we require the signal quality estimates to identify the corrupted regions. We use the previously estimated signal quality estimates to identify the corrupted regions.

4.5 Summary

We gave an overview of the signal quality indices available for ECG, ABP and PPG signals. Then, we introduced a novel morphological dissimilarity metric to estimate the signal quality of any quasiperiodic physiological signal. After that, we discussed the improvements on this method. Finally, we summarized the details of the implementation of the signal quality estimation in our framework.

Chapter 5

Reconstruction of Multi-parameter Physiological Signals

This chapter discusses the methods for reconstructing the corrupted regions in the physiological signals. Specifically, we detail the solutions to the subproblems listed in Section 1.4, feature representation, learning, and estimation. We introduce the problem in Section 5.1. In Section 5.2, we present the background, and discuss the related work. We begin Section 5.3 with an overview of the algorithm followed by a detailed discussion on the subcomponents of the algorithm. Section 5.4 gives the details of the experiments carried out, and the results. Finally, we conclude this chapter with the discussion and summary.

5.1 Introduction

We address the problem of identifying the corrupted regions in a multi-parameter signal, and reconstructing them using the information available in the correlated signals.

As in earlier chapters, we consider a multi-parameter signal represented by a matrix $\mathbf{S}_{n\times 2}$. It contains two correlated synchronized physiological signals. Each cell $s_{i,j}$ contains one sample. Samples may be corrupted in an unknown fashion. Our goal is to identify the corrupted regions, and estimate the actual sample values on

that region. In this chapter, we limit our focus to reconstructing the morphological features of the corrupted ECG signals. We do not consider the baseline and global trends in this chapter.

In our method, we use a database of templates. Here, a template is a segment of the multi-parameter signal that was chosen from previously seen regions free of signal corruption. When we come across the segments of high signal quality, we add them to the database; thus, we learn new morphologies.

The method is based on finding the closest match (template) to the corrupted segment from the database, time-warping the template to fit the corrupted segment's interval, and replacing the corrupted segment with it. The signal and the template contain two channels. The channel in the template corresponding to the corrupted channel in the signal is used for replacement, and the other channel in both the template and the signal is used for matching. The closest match is found using the DTW cost. As a preliminary step, we represent the segments by features to reduce the dimensions of the lookup entry, and preserve the clinically relevant information.

5.2 Related Work

There have been attempts to exploit the information available in the correlated channels of a multi-parameter physiological signal to assist automated medical systems to produce results that are more reliable. For example, researchers have tried to fuse information from various ECG channels, and other signals to robustly estimate the heart rate [1, 13, 38]. Further, the problem of reconstruction was the theme of the 11^{th} annual PhysioNet/CinC challenge [41].

5.3 Method

First, we present a quick overview of our method. A detailed description of the subparts is provided later. Bold capital letters represent a matrix $\mathbf{S}$, non-bold capital letters denote a column vector A, and lower-case letters denote a scalar w.

5.3.1 Overview

Goal : Let $\mathbf{S} \in \Re^{n \times 2}$ be a multi-parameter time series consisting of set of two single parameter physiological signals. The goal is to identify the corrupted regions $\{\mathbf{U}_i\}$, where $\mathbf{U}_i = \mathbf{S}_{[t_{i_1}, t_{i_2})}$, and reconstruct the corrupted samples in that region with the estimates $\{\mathbf{V}_i\}$ where $\mathbf{V}_i = \mathbf{S}*_{[t_{i_1}, t_{i_2})}$.

Procedure : First, we detrend the signal, and remove the baseline wander using a low pass filter[1]. Then, we extract the segments using the method described in Chapter 3. We run the algorithm starting at the first segment $\mathbf{U}_1$, continuously evaluating the signal qualities of the segments, and determining whether the segment needs reconstruction. If the segment needs reconstruction, we find the entry from the database that closely matches the current segment, and reconstruct the current segment using the signal segment in that entry. We also add new segments to our database if they are of high signal quality. This process is iterated over all the segments.

An iteration : We start each iteration with a segment $\mathbf{U}_i = \mathbf{S}_{[t_{i_1}, t_{i_2})}$ from $\mathbf{S}$, where the length of the segment is $\ell_i = t_{i_2} - t_{i_1}$. Using Morphological Dissimilarity, we determine whether the segment is corrupted, and whether it requires reconstruction.

If the signal needs reconstruction, i.e., if the Signal Quality Estimate (SQE) is below a threshold ($q_i < \zeta_{low}$), we proceed with the reconstruction process. First, we build the feature representation F_i of the segment. The signal $\mathbf{S}$ is a 2-parameter signal. Hence, $\mathbf{U}_i$ contains two correlated synchronized signals, and F_i is the joint representation of the both. We search the database, using F_i as the key, and find the top 20 matches. We find the best match on this set using the DTW distance between the clean channel of the segment $\mathbf{U}_i$ and the corresponding channel in the top matches. If the cost of the match is above a threshold, i.e., $c_i > \kappa$, we abort the reconstruction process on the current segment, and continue with the next segment. Otherwise, we use the best match as a template for reconstructing the corrupted signal. We time-warp the channel V_i^a from the matching template $\mathbf{V}_i$, with the clean channel U_i^a from the segment $\mathbf{U}_i$ to obtain the alignment w. Then, we replace the

[1] http://www.mit.edu/ gari/CODE/FILTERS/Baseline%20Wander%20Removal.htm

corrupted channel U_i^b of the current segment $\mathbf{U}_i$, with V_i^b* that is obtained from the template $\mathbf{V}_i$ by warping V_i^b using the alignment w.

If the signal doesn't need reconstruction, i.e, if the SQE is above a threshold ($q_i > \zeta_{high}$), we try to learn the morphology from that segment. First, we build the feature representation F_i of the segment, and add the entry ($\mathbf{F}_i \Rightarrow \mathbf{U}_i$) to the database.

5.3.2 A segment

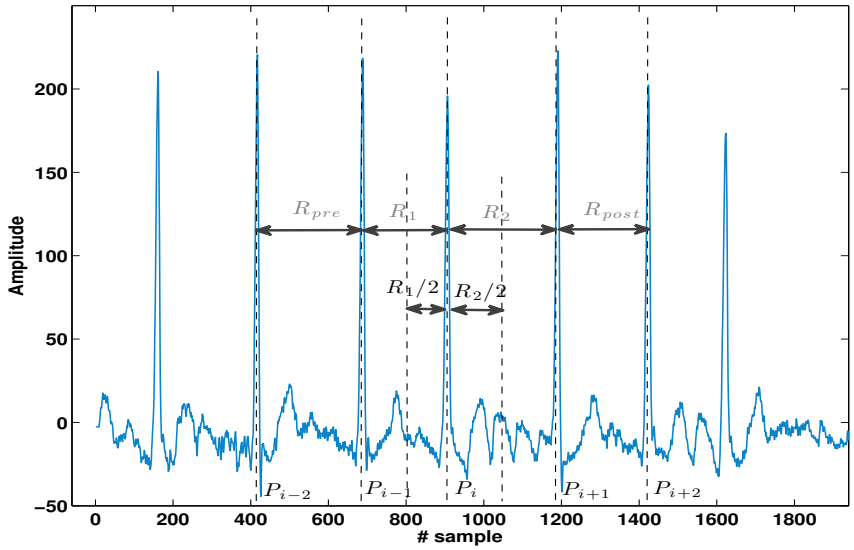
A segment for reconstruction is the unit of the signal from the mid point of one R-R interval to the mid point of the next R-R interval.

The template we use treats the R-R interval a segment. Since the rate of change is the highest at R peaks, this choice allows us to identify the segment boundaries with the highest resolution, i.e., up to single sample. Though using R-R intervals as segments is common, it is not appropriate for this application. First, we want a smooth transition between segments, which suggests that segment boundaries should occur in places where the signal is changing slowly. Secondly, since the shape of the QRS complex is important in many applications, we need to reconstruct it accurately. This is easier if the entire complex lies within a single segment. For this reason, we use the R peaks as the center of segments rather than as the border.

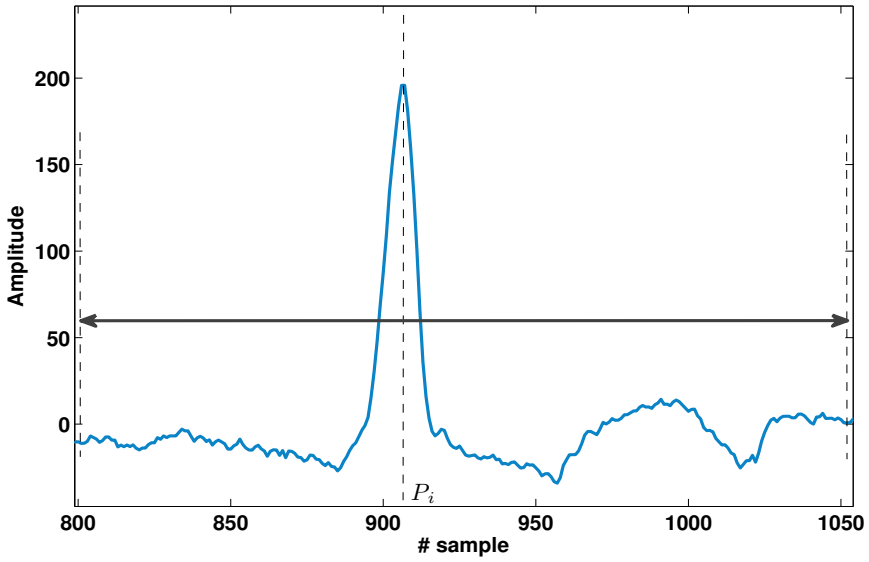
We start with the segment boundaries $\{p_i\}$ obtained using the joint segmentation (Section 3), and construct the segments, as illustrated by Figure 5-1, so that they are centered on R-peak p_i.

5.3.3 Identifying corrupted regions

We reconstruct only the corrupted regions, i.e., those with a poor SQE. First, we use the SQE obtained from Morphological Dissimilarity (Section 4) to identify the corrupted regions. Based on the SQE values, there are three possibilities (Figure 5-2). If neither signal appears to be corrupted, we update the templates, but make no change to the signals. If the SQE for exactly one signal in the segment falls below a



(a) Segment extraction and common features



(b) One Segment

Figure 5-1: Figure (a) illustrates how a segment is extracted from the signal **S** using the segment boundaries $\{p_i\}$ obtained in Section 3. While extracting the segment, the features that depend on R-R lengths (R_{pre}, R_1, R_2, R_{post}) are also constructed. Figure (b) defines one segment. The i^{th} segment U_i is centered around p_i.

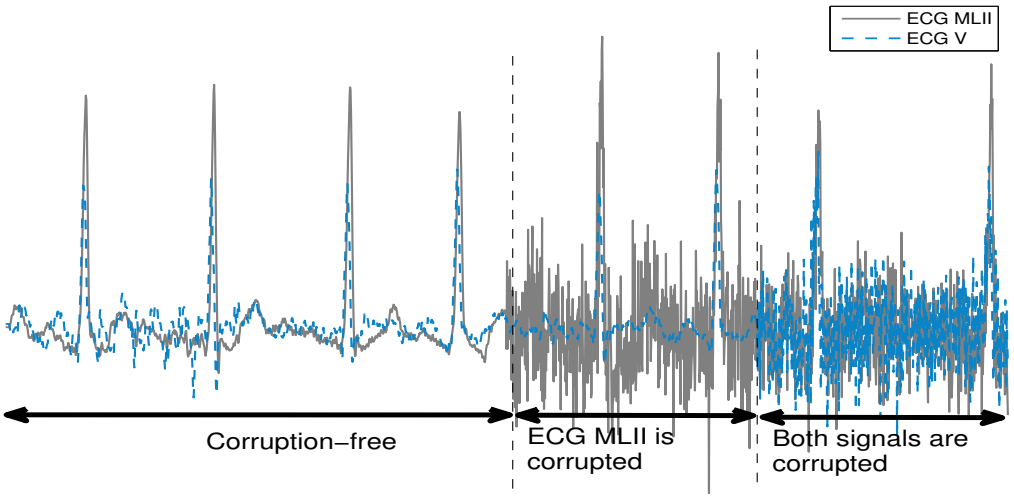


Figure 5-2: An example distribution of the corruption. When one signal is corrupted we try to reconstruct them. When both signals are corrupted, the reconstruction process only flags that segment, but does not try to reconstruct it.

threshold, we try to reconstruct it. Since the correlated signal is used to estimate the sample values of the corrupted signal, when both signals are corrupted, we don't have enough information left in the signal to reconstruct them. Therefore, if the corruption spans both the signals, we abort the process, and continue with the next segment. In such case, we flag that segment, so that any systems that depend on these signals could avoid triggering false alarms.

5.3.4 Feature Representation

In reconstruction, we search the database for the closest match to the current segment. Since we want to do it in real-time with a growing database, we need a fast method of retrieval that could scale. A method with linear time complexity ($O(\ell)$) for comparison between two sequences of length ℓ, and sub-linear time complexity for lookup on database of size n is highly desirable. In addition, we also want the method to provide a level of abstraction that increases the probability of finding a physiologically relevant match.

Since the segments are usually of different lengths, a direct comparison function,

such as the Euclidean distance is not suitable. On the other hand, variable length metrics such as DTW, LCSS, and Edit Distance On Real Sequence (EDR) are of quadratic time complexity. Further, lookups using these methods for comparison do not easily scale with large databases. We, therefore, resort to an intermediate representation of the segments using features.

Feature representation helps us represent a segment with a fixed length vector, hence two sequences can be compared in linear time in feature space. Further, it provides the level of abstraction desired, thus helps us generalize. The feature representation also helps our method preserve the clinically relevant information such as PVCs. Two beats with Premature Ventricular Contraction (PVC) could differ in morphology, but they might still indicate the same clinical event. The set of features we chose helps us represent this information better, and avoid over-fitting. Hence, it is used to find the set of closest matches from the databases to an entry, loosely serving as a hash-code.

Every segment $\mathbf{U}_i$ contains two correlated synchronized signals, and F_i is the joint representation of the both. Table 5.1 lists the set of features in the feature representation $F_i = \{f\}$. First four rows of the table contain the features related to R-R intervals, followed by the features of the signal in the segment.

5.3.5 Dimensionality Reduction

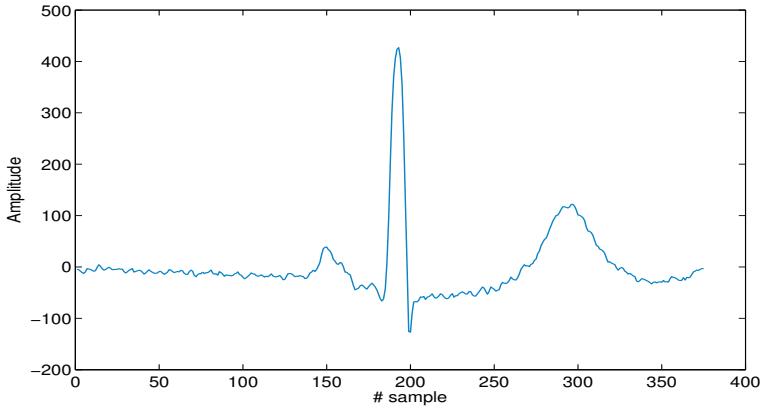
The database may contain thousands of entries. Because signal in each record is of varying length, we use the DTW distance to measure the similarity. The computation of the DTW distance has a quadratic time complexity. Therefore, to lookup an entry, a pairwise comparison with all the records in the database can be unacceptably slow.

We speed up the lookup by reducing the dimensionality of the search. We represent the signal with the features, and use the features to do similarity search. For a fixed length feature vector, the comparison can be done in time that is sub-linear on the size of the database. We achieve further speed up through other optimizations discussed in Chapter 7.

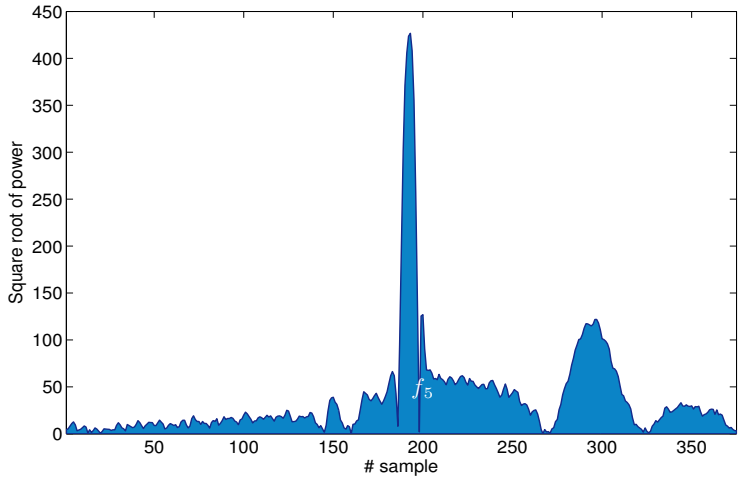
The top K matches are found through k-nearest neighbors search (KNN) with a

Table 5.1: Set of features that are used to represent a segment.

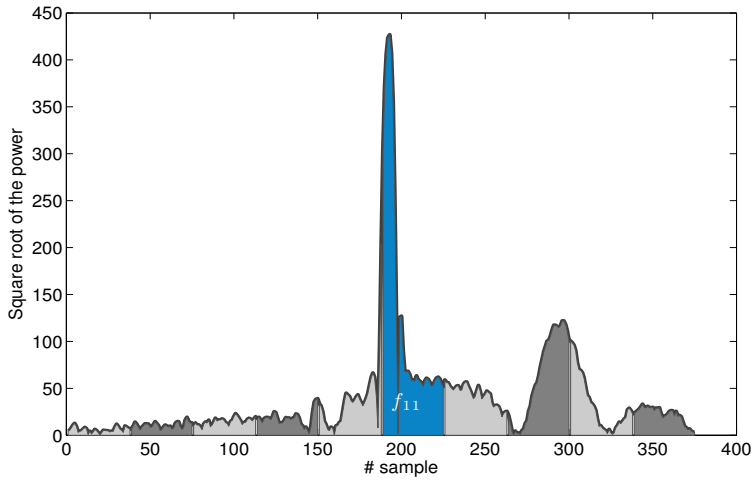
ID	Feature	Description
f_1	R_{pre}	The length of the previous R-R interval (Figure 5-1)
f_2	R_1	The length of the R-R interval constituting the first part of the segment (Figure 5-1)
f_3	R_2	The length of the R-R interval constituting the latter part of the segment (Figure 5-1)
f_4	R_{post}	The length of the next R-R interval (Figure 5-1)
f_5	E_{total}	Square root of the total energy of the first signal in the current segment (Figure 5-3)
f_6 .. f_{15}	E_k	The fraction of the square root of the energy in the k^{th} section of the first signal in the segment (Figure 5-3)
f_{16}	K	Kurtosis of the sample values of first signal in the segment
f_{17}	D_{total}	DTW distance between the first signal in the segment and the median of the same signal
f_{18} .. f_{27}	D_k	DTW distance between the k^{th} subsequence of the first signal in the segment and the k^{th} subsequence of the median of the same signal
f_{28}	SDR	Fraction of spectral energy in the QRS complex of the first signal in the segment
f_{29}	MAX	The maximum sample value of the first signal in the segment
f_{30}	MIN	The minimum sample value of the first signal in the segment



(a) The signal from a segment



(b) f_5 : Square root of the total energy of the signal



(c) f_6 .. f_{15} : The fraction of the square root of the energy in the k^{th} section of the signal

Figure 5-3: Estimation of features f_5 (b), and f_{11} (c) on an ECG signal segment(a).

normalized euclidean distance function.

$$d(\vec{X}, \vec{X}_i) = (\vec{X} - \vec{X}_i)^T(\vec{X} - \vec{X}_i) \quad (5.1)$$

$$K = \underset{i \in N_X}{\operatorname{argmin}}\ d(\vec{X}, \vec{X}_i) \quad (5.2)$$

Here, N_X is the set of neighbors for the entry $\vec{X}$.

We select the top twenty matches, and from them, we find the candidate for replacement V_i using both the DTW distance and the feature distance.

$$V_i = \underset{k}{argmin}\Big\{ \sum_j (F_{k,j} - F_{i,j})^2 + \lambda.dtw(U_k^a, U_i^a) \Big\} \quad (5.3)$$

$$c_i = \underset{k}{min}\Big\{ \sum_j (F_{k,j} - F_{i,j})^2 + \lambda.dtw(U_k^a, U_i^a) \Big\} \quad (5.4)$$

Here, F_k and F_i are the feature vectors of the segments (top 20 matches) $\{\mathbf{U}_k\}$, and the current segment $\mathbf{U}_i$ respectively. The summation is performed over all the features in the feature vector. The DTW distance is computed only between the clean channels of the segments (U_k^a and U_i^a). The regularizer $\lambda = 25$ was empirically chosen. The combination helps us avoid over-fitting, and balance the preservation of clinically relevant events against the morphological similarity.

5.3.6 Reconstruction

We want to reconstruct the corrupted channel of the current segment U_i^b with the corresponding channel from the replacement candidate V_i^b that was found.

We first verify the correctness of the match found. We accept the reconstruction only if the cost of the match c_i (Equation 5.4) is less than a threshold. If the cost c_i is greater, we flag the segment $\mathbf{U}_i$ so that automated systems could avoid producing false alarms in those regions.

Since the length of the current segment $\mathbf{U}_i$ and the length of the candidate found (template) $\mathbf{V}_i$ are typically unequal, we next time-warp the template with the current segment. Time-warping is done by finding the optimal alignment $\phi(k)$ between

the clean channel of the current segment U_i^a and the corresponding channel of the template V_i^a (Equation 5.5-5.7).

$$\phi(k) = (\phi_1(k), \phi_2(k)), 1 \leq k \leq K \tag{5.5}$$

$$C(V_i^a, U_i^a) = \min_{\phi} C_\phi(V_i^a, U_i^a) \tag{5.6}$$

$$C_\phi(V_i^a, U_i^a) = \sum_{k=1}^{K} d(V_i^a[\phi_1(k)], U_i^a[\phi_2(k)]) \tag{5.7}$$

We then replace each sample of the corrupted channel $U_i^b[x]$ with the time-warped sample $V_i^b[x*]$, which is obtained from the median of the samples with which it is aligned.

$$x* = median(\phi_2(k)), 1 \leq k \leq K \text{ and } \phi_1(k) = x \tag{5.8}$$

5.4 Experimental Results

In this section, we present the experimental results for our method. We begin with the description of our dataset, and then present the results of the experiments.

In our experiments, we use the multi-parameter ECG data from MIT-BIH Arrhythmia Database at Physionet.org [21]. The database has 48 ECG waveform records, each contains two channels and is 30 minutes long. The recordings are digitized at 360 Hz with 11-bit resolution over 10mV range. The recordings were selected to include less common but clinically significant arrhythmias. This helps us evaluate the robustness of our method.

We add synthetic corruption that is intended to mimic real-world corruption to the data (selected channel of a record). We evaluate our method by quantifying the effectiveness of the reconstruction on this corrupted data. We compare the reconstructed data, and the corrupted data with the original data to evaluate the performance of our method.

We used only 39 records from this set. Records 100-108 were excluded from the experiments, because they contain significant amount of noise and transient corrup-

tion on both channels. We would not able to use those records effectively, because we require a gold standard to compare against, and the second channel to be corruption free so that it can be used to reconstruct the artificially corrupted first channel. Other records might also contain sporadic transient corruptions. Although it is unfavorable to our test cases, we do not exclude any region in these 39 records, even when both channels are corrupted.

We use the following criteria for comparison.

1. **Q1 : Residual distance :** We measure the similarity between the reconstructed data (S^b*), and the original uncorrupted data (S^b) by measuring the Euclidean residual distance r of the reconstructed data.

$$r = \sqrt{\frac{\sum_k^n (S^b * [k] - S^b[k])^2}{n \times \sigma_S^2}} \tag{5.9}$$

   We normalize the Euclidean distance to make it comparable across the records.

2. **Q2 : Reproducibility :** Our ultimate goal is to enable the automated analysis systems produce more reliable results. Hence, we test our method's ability to improve the classification accuracy of a common task. We run a widely used Premature Ventricular Contraction (PVC) detector[2] on the original data (S^b), the artificially corrupted data ($S^b\#$) and the reconstructed data (S^b*), and record their agreements. If the PVCs are detected within 150 ms on two signals, we consider it an agreement. We quantify the ability to preserve the clinically relevant events by counting the disagreements (Algorithm 1). The number of disagreement $n_{\text{disagreement}}$ is evaluated between the original data (S^b), and the artificially corrupted data ($S^b\#$), and between the original data (S^b), and the reconstructed data (S^b*). The disagreement Δ is finally expressed in terms of the fraction between the total number of disagreements $n_{\text{disagreement}}$, and the total number of beats n_{beats} in the region.

$$\Delta = n_{\text{disagreement}}/n_{\text{beats}} \tag{5.10}$$

[2]http://www.eplimited.com/software.htm

Algorithm 1 Disagreement

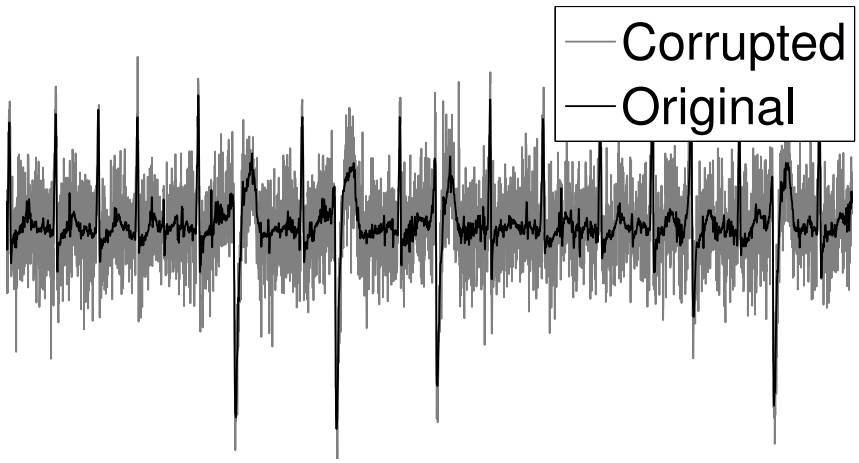
$n_{\text{disagreement}} = 0$
for all beat in beats **do**
  if there is PVC in S_1 **then**
    if there is no PVC in S_2 **then**
      $n_{\text{disagreement}} = n_{\text{disagreement}} + 1$
    end if
  else if there is PVC in S_2 **then**
    if there is no pVC in S_1 **then**
      $n_{\text{disagreement}} = n_{\text{disagreement}} + 1$
    end if
  end if
end for

5.4.1 Experiment 1 : Effectiveness of Reconstruction

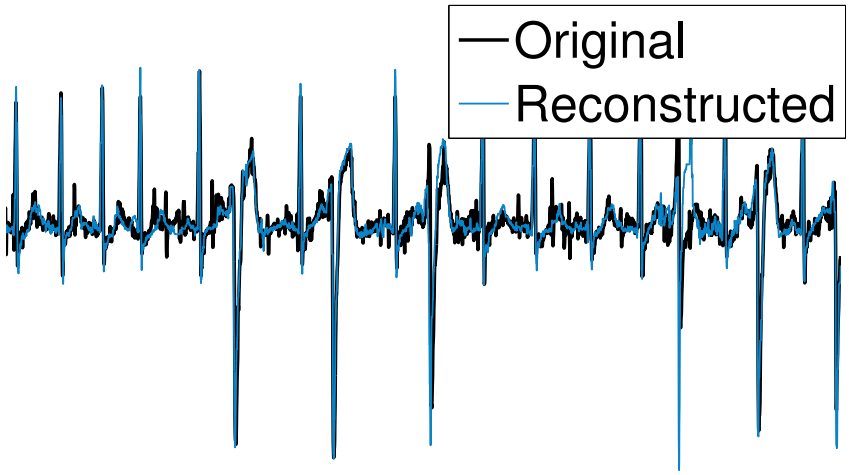
We corrupt the first channel of the records with the additive white gaussian noise (AWGN) at 0dB SNR. We add the artificial corruption to only the last 20% of the data, and attempt to reconstruct it. We build the database from the first 80% of the data.

Table 5.2 summarizes the results. It shows that our method reduces the residual distance (Q_1) by 250% for a signal corrupted at SNR 0dB. Further, on average, it was able to improve the classification accuracy (Q_2) by more than seven fold. Figure 5-4, shows the effectiveness of the reconstruction on Record 200.

Table 5.3 lists the performance of our method on each record. The first three columns describe the record, the next two columns provide the results for the Q_1 criterion on both corrupted and reconstructed channels, and the last six columns provide the results for the Q_2 criterion on both corrupted and reconstructed channels. Q_1 is measured by the normalized residual distance, and Q_2 is dependent on the total number of disagreements, which is the sum of misses and inventions. A miss is counted, when the PVC found on the original signal S^b is missed in $S^b\#$ or S^*. An invention is counted, when a new PVC is found in $S^b\#$ or S^*. The reconstruction not only improves the morphological similarity, but also preserves the clinically relevant information. On some records, e.g., 207 and record 217, our method performs poorly.



(a) Corrupted Vs Original



(b) Reconstructed Vs Original

Figure 5-4: Record 200 from MIT-BIH Arrhythmia database. The first channel is corrupted with AWGN at SNR 0dB (a), and reconstructed using our method (b).

This is mainly due to the poor quality of the correlated channel (channel 2). On record 217 in particular, channel 2 is extremely noisy, and at times completely corrupted.

Table 5.2: Experiment 1 : Summary

		Q1		Q2	
	PVC	$r_{S^b\#}$	r_{S^b*}	$\Delta_{S^b\#}$	Δ_{S^b}
Median	4	1.01	0.39	0.09	0
Average	47.64	1.01	0.40	0.14	0.02

5.4.2 Experiment 2 : Different SNR Levels

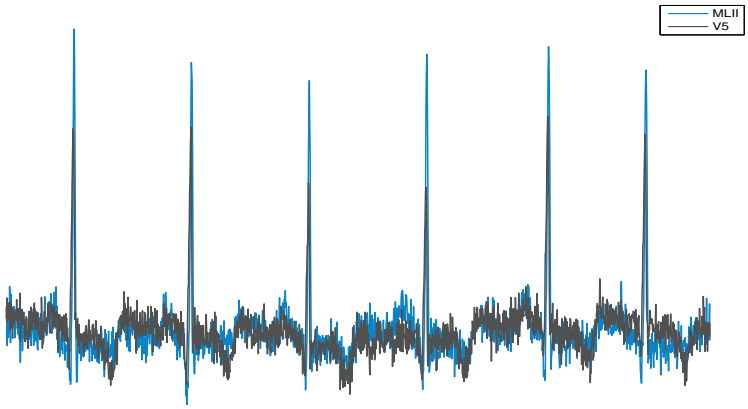
On all 39 records, we corrupt the last 20% of the first channel with AWGN at SNR levels of 10 dB, 0 dB, and -10 dB (Figure 5-5). We build the database from the first 80% of the data. For the remainder, we check whether the segment is corrupted and reconstruct the segment if it is corrupted.

Table 5.4 summarizes the average disagreement (Δ), and the residual distance (r) for the reconstructed signal (S^b*). Somewhat surprisingly, we get the worst performance at the highest signal to noise ratio. In addition, at low SNR levels, the performance does not deteriorate with decreasing signal quality.

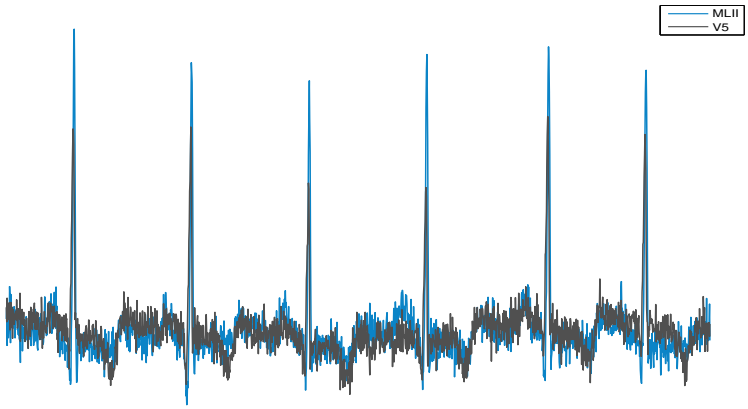
Our method makes a binary decision to either reconstruct the corrupted region, or leave it. It does not support partial recovery of the signal. This is both the strength and the weakness of the algorithm. By design, once the region is determined to be corrupted, the algorithm will choose to reconstruct it completely using the correlated channel and the history of the signal. Therefore, at low SNR levels, the level of corruption does not affect the performance of the reconstruction process. In addition, at higher SNR levels, when the signal is of high quality, the algorithm will choose not to reconstruct the signal, and, hence the original performance will be retained. However, in the middle, for instance, at SNR 10dB, corrupted regions are not easily distinguishable, and hence it results in poorer performance compared to that of lower signal qualities.

Table 5.3: Experiment 1 : Effect of reconstruction

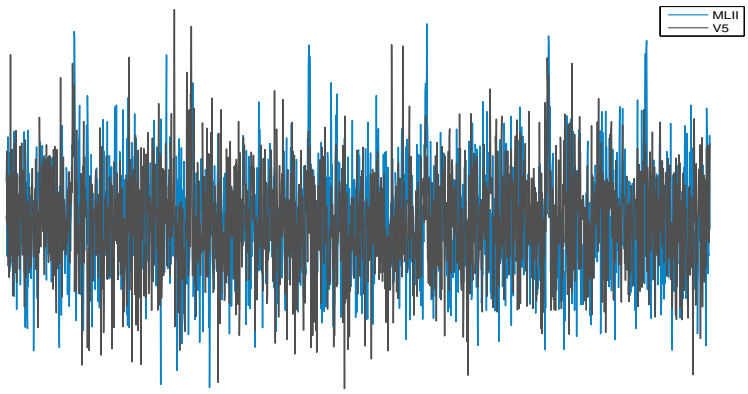
	In last 6 minutes		Q1		Q2-unreconstructed			Q2- reconstructed		
Record	Beats	PVC	$r_{S^b\#}$	r_{S^b*}	miss	invention	Δ	miss	invention	Δ
ID	#	#			#	#		#	#	
109	504	6	1.33	0.59	27	3	0.06	0	0	0
111	424	1	0.93	0.50	25	1	0.06	1	1	0
112	507	0	0.95	0.30	33	0	0.07	0	0	0
113	356	0	1.02	0.28	32	0	0.09	0	0	0
114	375	0	1	0.66	38	0	0.10	1	0	0
115	389	0	0.98	0.47	34	0	0.09	0	0	0
116	481	11	0.60	0.42	36	9	0.09	0	1	0
117	305	0	0.93	0.16	24	0	0.08	0	0	0
118	457	4	1.07	0.40	31	4	0.08	1	0	0
119	416	87	0.98	0.19	32	66	0.24	0	0	0
121	372	0	0.97	0.55	36	0	0.10	0	0	0
122	493	0	1.01	0.15	51	0	0.10	0	0	0
123	301	2	1.02	0.21	22	2	0.08	0	0	0
124	324	3	0.96	0.30	18	3	0.06	0	0	0
200	555	149	0.94	0.44	23	120	0.26	8	7	0.03
201	405	23	1.08	0.42	21	18	0.10	0	7	0.02
202	426	1	1.16	0.41	38	1	0.09	0	0	0
203	619	71	0.94	0.80	35	61	0.16	11	41	0.08
205	531	36	1.04	0.49	42	31	0.14	0	8	0.02
207	473	224	1	0.73	5	170	0.37	4	131	0.29
208	605	235	0.98	0.44	15	187	0.33	3	3	0.01
209	607	2	1	0.61	41	2	0.07	0	2	0
210	534	31	1.02	0.38	41	25	0.12	10	4	0.03
212	549	0	0.97	0.30	32	0	0.06	0	0	0
213	655	53	0.98	0.26	35	42	0.12	3	2	0.01
214	456	45	0.99	0.42	40	34	0.16	2	4	0.01
215	676	40	1.03	0.39	52	32	0.12	2	1	0
217	453	436	1.09	0.37	0	374	0.83	0	116	0.26
219	459	11	1.03	0.21	30	7	0.08	0	0	0
220	411	0	1.02	0.29	17	0	0.04	0	0	0
221	489	32	0.89	0.29	39	24	0.13	0	0	0
222	523	2	1.08	0.76	37	2	0.07	2	2	0.01
223	525	97	1.09	0.33	29	74	0.20	2	5	0.01
228	425	74	1.02	0.39	20	59	0.19	0	1	0
230	490	1	1.12	0.50	35	1	0.07	0	0	0
231	399	0	1.06	0.32	10	0	0.03	0	0	0
232	361	0	1.03	0.34	23	0	0.06	0	0	0
233	627	180	1	0.32	16	141	0.25	0	6	0.01
234	550	1	1.04	0.29	28	0	0.05	0	0	0



(a) SNR 10dB



(b) SNR 0dB



(c) SNR -10dB

Figure 5-5: Record 123, from MIT-BIH Arrhythmia database. The original signal is added with AWGN at SNR = 10dB, 0dB and -10dB.

Table 5.4: Experiment 2 : Effect of SNR levels

	Similarity : r_{S^b*}	Disagreement : Δ_{S^b}
10	0.410	0.031
0	0.401	0.021
-10	0.402	0.022

However, on all three noise noise levels, our method improved the classification accuracy, as measured by the disagreement, by more than 460%, and the increased the similarity to the original signal, as measured by the normalized residual distance, by more than 200%.

5.4.3 Experiment 3 : Simulated Real-World Corruptions

Table 5.5: Performance against different types of real-world corruptions

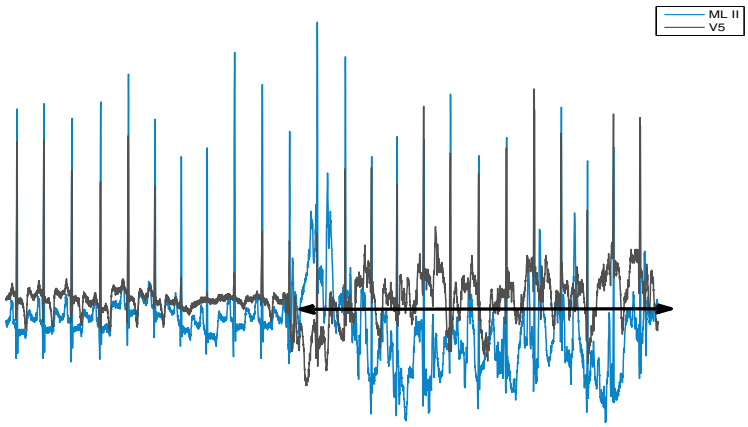
	Similarity : r_{S^b*}	Disagreement : Δ_{S^b}
AWGN	0.410	0.031
EM	0.36	0.023
MA	0.19	0.003
BW	0.05	0.001

We alter the first 20% of the first channel with the following types of corruptions at SNR = 10 dB: Additive White Gaussian Noise (AWGN), Electromagnetic Interference (EM), Muscle Artifact (MA), and Baseline Wander (BW) (Figure 5-6). We use MIT-BIH Noise Stress Test Database[3] and *nstdbgen*[4] to generate the first three types of noise models. We chose to corrupt at SNR 10dB, because at lower signal qualities, the distinction between different noise models become less apparent and signals with high signal quality does not require reconstruction.

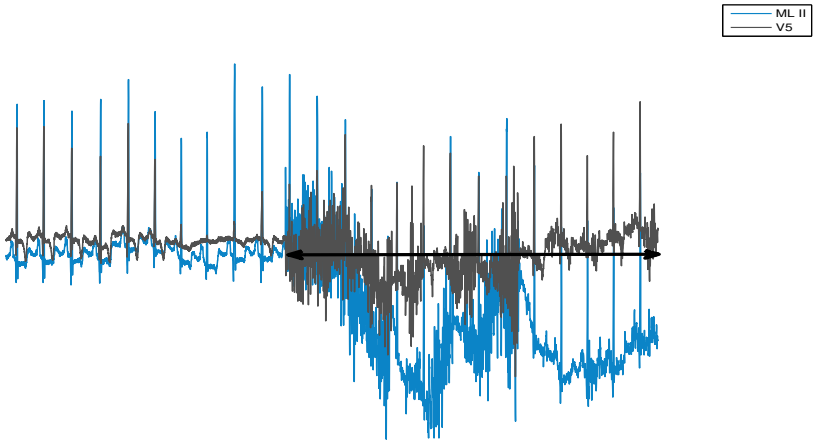
Table 5.5 summarizes the average disagreement (Δ), and the residual distance (r) for the reconstructed signal (S^b*). We achieve the best performance for Baseline

[3]http://www.physionet.org/physiobank/database/nstdb

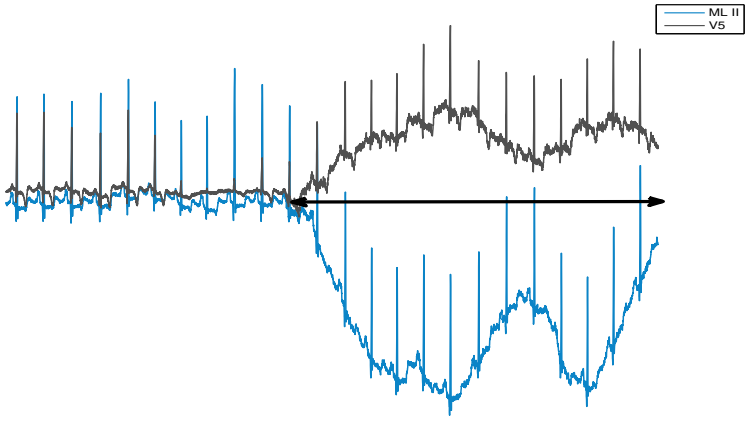
[4]http://www.physionet.org/physiobank/database/nstdb/nstdbgen-



(a) Electromagnetic Interference (EM)



(b) Muscle Artifact (MA)



(c) Baseline Wander (BW)

Figure 5-6: Record 123, from MIT-BIH Arrhythmia database. The original signal is added with the following types of corruptions at SNR 10dB : Electromagnetic Interference (EM), Muscle Artifact (MA), and Baseline Wander (BW). The highlighted area indicates the corrupted region.

Wander. In preprocessing, we remove baseline wander using a low-pass filter[5]. The baseline wander removal algorithm was effective at 10 dB SNR, and was able to cancel the noise itself. The worst performance was observed for AWGN, and EM noise. Hence, on all four noise noise models, our method improved the classification accuracy, as measured by the disagreement, by more than 500%, and the increased the similarity to the original signal, as measured by the normalized residual distance, by more than 250%. Further, it must be noted that this is the worst case performance, and BW and MA are more common than EM and AWGN in practice.

5.4.4 Experiment 4 : Size of the Database

In this experiment, we test the influence of the length of the database and learning. We corrupt the first channel of the records with AWGN at 0dB SNR. We corrupt the last 20%, 50%, and 80% of the first channel of the signal. Therefore, the length of signal available to build the database shrinks from 80% to 20% of the original signal. For the remainder, we check whether the segment is corrupted, at the granularity of a beat, and reconstruct the segment if it is corrupted.

Figure 5-7 summarizes the average disagreement (Δ), which is the measure of reproducibility, and the residual distance (r) for the reconstructed signal (S^b*), which is the measure of the similarity. Both metrics worsen with increasing length of corrupted regions. The average disagreement rises exponentially. There are two contributing factors for this observation. First, we are unable to reconstruct the PVC beats properly with shrinking database. Another reason is that with larger corrupted regions, we are estimating the sample values farther from where we learned the relationships.

5.4.5 Experiment 5 : Learning

We corrupt three randomly chosen non-overlapping, 5 minute long windows (50% of the signal) starting from the 6^{th} minute (20% of data) on the first channel of the records with AWGN at 0dB SNR. We build our database using the first 20% of

[5]http://www.mit.edu/ gari/CODE/FILTERS

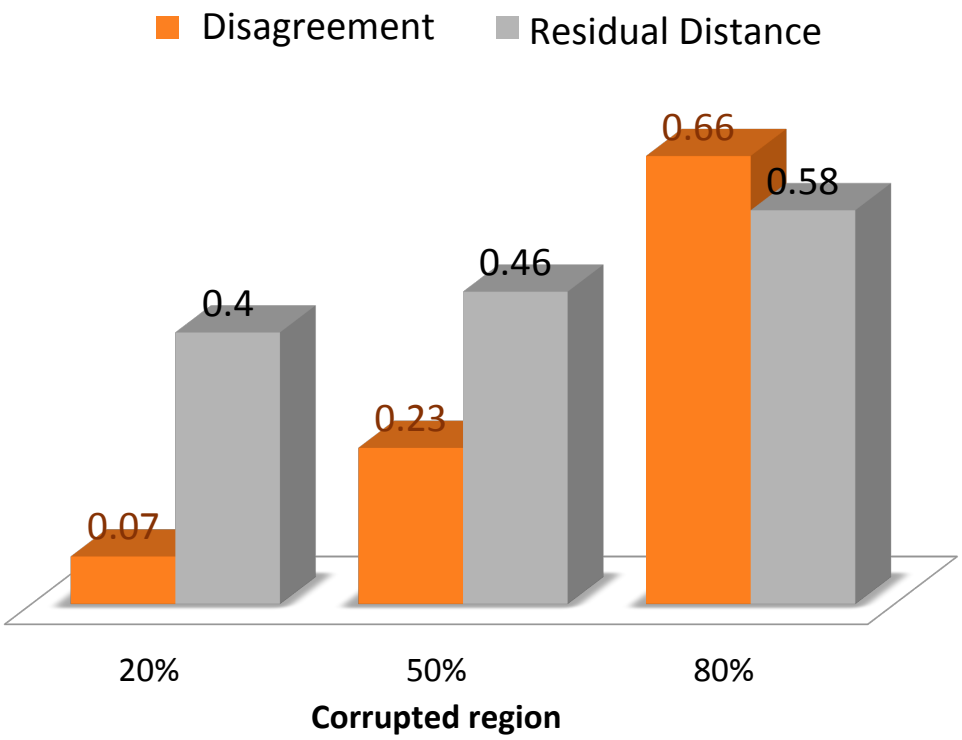


Figure 5-7: Last 20%, 50%, and 80% of the first channel of the signal are corrupted with AWGN at 0dB SNR. The average disagreement (Δ) and the residual distance (r) are computed for all 39 records.

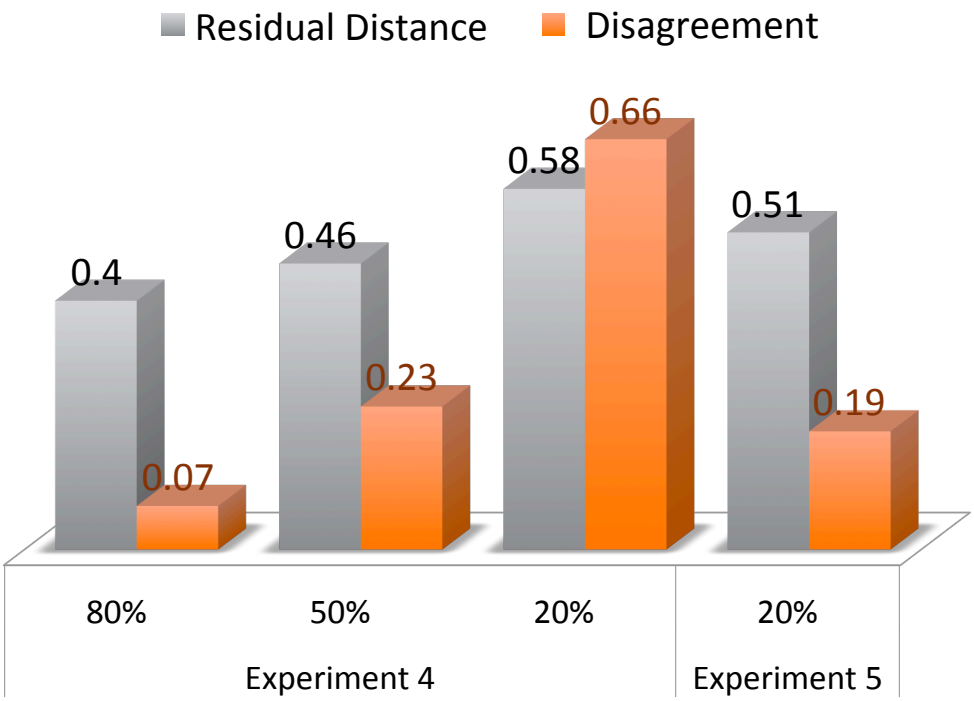


Figure 5-8: In experiment 4, we build the database using the first 20%, 50%, and 80% of the signals. The remainder of the signals is corrupted with AWGN at SNR 0dB. The average disagreement (Δ) and the residual distance (r) are computed for all 39 records. In experiment 5, we build the database using the first 20% of the signals. On the remainder, three randomly chosen non-overlapping regions (each 5 minute long), are corrupted with AWGN at SNR 0dB. Hence, totally 50% of the data is corrupted. Here, we also learn the new morphologies over the regions free of corruption.

the signal. For the remainder, we check whether the segment is corrupted, at the granularity of a beat, and reconstruct the segment if it is corrupted. If the segment is determined to be free of corruption in both channels, and is of high signal quality, we add that to the database, thus learn the new morphologies.

Figure 5-8 compares the results to that of Experiment 4, and demonstrates the benefits of learning new relationships and morphologies on the fly. Though, we start with a database built from only 20% of the signal, we are able to achieve the performance similar to that, where we start with a database generated from 50% of the signal. The difference in residual distance is not large. However, there is a significant improvement in disagreement. We attribute this to adding new segments with PVCs to the database, and being able to learn the morphologies closer to the place of reconstruction.

5.5 Summary

In this chapter, we presented a method for reconstructing a corrupted signal in a multi-parameter physiological signal using the information available in a correlated signal. The method uses template matching to reconstruct the signal. The best match is found from the database of templates using a combination of DTW cost and the feature distance.

Using the data from the MIT-BIH Arrhythmia Database, we conducted a series of experiments to test the effectiveness of our method. We added synthetic corruption to the data, and used this artificially corrupted data to evaluate our method. We quantify the effectiveness of the reconstruction by comparing the reconstructed data, and the corrupted data with the original data. Our evaluation criteria were normalized residual distance and classification accuracy.

On 39 records from MIT-BIH Arrhythmia Database, our method improved the classification accuracy by more than 700%, and increased the similarity to the original signal, as measured by the normalized residual distance, by more than 250%. We also compared the performance for various types of real-world corruptions, and different

SNR levels. Finally, we showed the effectiveness of dynamically determining the signal quality, at the granularity of a beat, and learning the signal morphology on the fly.

Our method is particularly useful when the system suffers noise and transient corruption. We used only ECG signals in this chapter, because we were able to obtain the database containing handpicked, less common but clinically significant events only for ECG. However, we believe that our method is applicable to any set of correlated physiological signals such as d ABP, PPG, and CVP as well.

Chapter 6

Data Fusion

This chapter discusses the details of fusing estimations from multiple signals. We begin Section 6.1 with details of the method. In Section 6.2, we discuss the methods to handle the baseline. In Section 6.3, we discuss the measures of performance used to evaluate our method, and present the results of a series of tests in which comparisons are made using each of the performance measures. Finally, in Section 6.4, we summarize our work.

6.1 Data Fusion

Goal : Let $\mathbf{S} \in \Re^{n \times m}$ be a multi-parameter time series consisting of set of m single parameter physiological signals. The goal is to find the corrupted regions in signal S^1, and reconstruct it using estimations obtained separately from the other correlated signals S^j from $\mathbf{S}$, where $j = 2, 3..m$.

Procedure : First, we preprocess the signals. Then we jointly segment them, and estimate the SQE of each segment for each of the signals. Using the segment boundaries, we divide the signal into quasiperiodic units, which serve as the units of replacement. We build a database using the segments of signal quality greater than a threshold. The database is built for each pair of signals: a signal that has corruptions and a correlated signal. Then we run the algorithm presented in Section 5.3, starting at the first segment U_1, continuously evaluating the signal qualities of the segments,

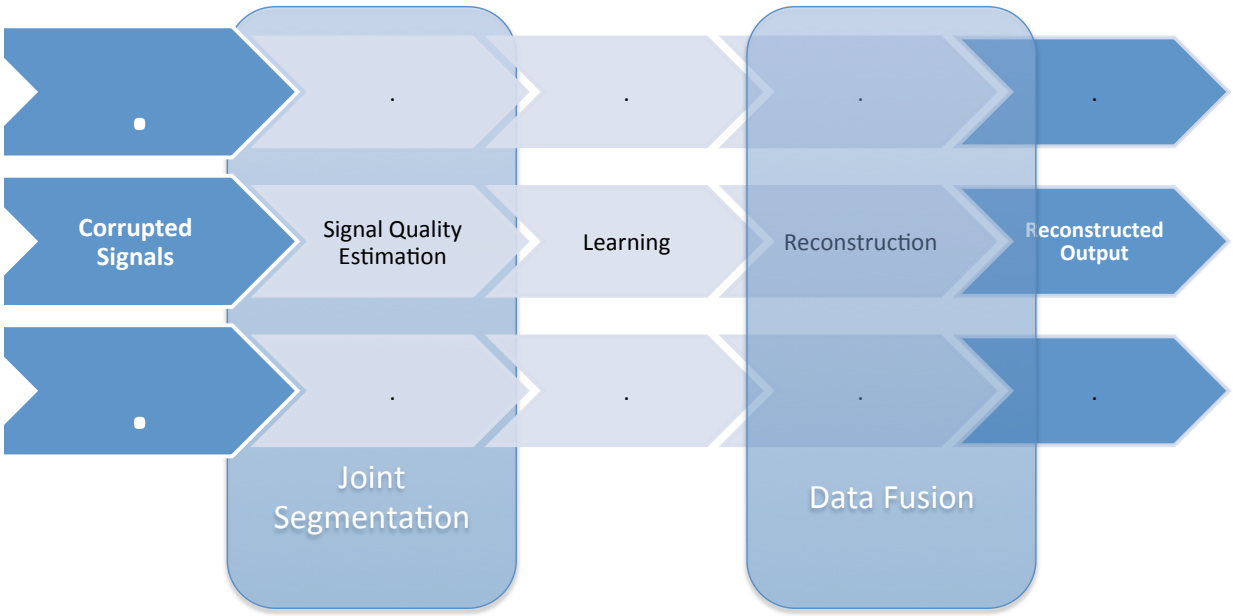


Figure 6-1: The picture outlines the reconstruction process on a multi-parameter timeseries. The multi-parameter signal contains several correlated physiological signals such as ECG, ABP, and PPG. First, we simultaneously perform the joint segmentation and the Signal Quality estimation on the multi-parameter signal. Then we learn the pair wise relationship between the signal-to-be-reconstructed and another correlated signal in the multi-parameter signal and build the database. This is repeated for all the correlated signals. We find the best matches separately for each correlated signal and finally combine them.

and determining whether each segment needs reconstruction. If the segment needs reconstruction, for each correlated signal, we find an entry from the corresponding database that closely matches the current segment using the method explained in Section 5.3.5. We weight each entry by the quality of reconstruction, and combine them. Figure 6-1 summarizes the complete process.

For the segment U_i^1, let $\{V_i^j | j = 2, 3..m\}$ be the matches found, and $\{c^j | j = 2, 3..m\}$ be the costs of the matches. The reconstruction V_i is obtained by fusing the matches,

$$V_i = \frac{\sum_{j=2}^{m} V^j (w^j)^p}{\sum_{j=2}^{m} (w^j)^p} \tag{6.1}$$

$$w^j = c^j . q_i^j . r^j . \tag{6.2}$$

Here, q_i^j is the SQE of the i^{th} segment of j^{th} signal, and r^j is the correlation coefficient between the signals (corrupted signal and j^{th} correlated signal) estimated around the segment V_i^j.

The choice of p influences the performance of our method. For instance, when the correlation between the corrupted signal and a correlated signal is high (e.g., two ECG channels), $p \to \infty$ tends to perform better. This is similar to selecting the most useful signal for the reconstruction of the segment U_i,

$$k = \underset{j=2,3..m}{\text{argmax}}\ w^j \tag{6.3}$$

$$V_i = V^k . \tag{6.4}$$

However, when the correlation between the corrupted signal and the correlated signals is low (e.g., an ECG channel and an ABP signal), lowering p to one seems to improve the performance.

6.2 Global Trend

In our analysis, we first remove the global trends and constant offsets. In ECG signals, the baseline represents noise, and it can be removed without affecting the analysis. Further, removing the trend negates the effects of baseline wander, and improves the performance of the automated analysis. However, signals such as ABP, and PPG contain their primary information in the baseline. Hence, our framework must accommodate the baseline of these signals.

In our framework, we first extract the baseline using a first order zero-phase lowpass filter. For ECG signals the $-3dB$ cutoff frequency is set at $0.2Hz$, and for ABP and PPG it is set at $0.07Hz$. Then we subtract the baseline from the signal, and reconstruct the corrupted segments on this baseline-subtracted signal. We reconstruct the baselines independently, and add them to the signals. Thus, we modularize the reconstruction into two subproblems :

1. Reconstruction of the baseline-subtracted signals. This is a quasiperiodic multi-parameter signal, and hence we focus on reconstructing the repetitive morphological structures in the signal. We discussed the methods to do this in Chapter 5 and Section 6.1.

2. Prediction of the global trends. Using the baselines of the multi-parameter signal, we try to estimate the values of the baseline of the corrupted signal over the segments of low signal quality.

The baseline of a multi-parameter signal is a multi-parameter time-series. We treat the reconstruction of the baseline as a regression problem. We take the mean of the samples on a segment as a sample point in our regression. Thus we model the relationship between a scalar variable, the mean of the samples of the segment from the signal that is to be reconstructed, and one or more variables, the means of the samples of the segments from the correlated signals. For instance, if the first signal S^1 in the multi-parameter signal $\mathbf{S}_{n\times m} = \{S^1, S^2..S^m\}$ contains corruption, the data

$\mathbf{X}_{n\times(m-1)}, Y_{n\times 1}$ for the regression is,

$$
\begin{aligned}
x_i^{(j-1)} &= mean(S^j_{[p_i,p_{i+1})}) \forall j = 2, ..m && (6.5)\\
y_i &= mean(S^1_{[p_i,p_{i+1})}) && (6.6)\\
\mathbf{X} &= \begin{bmatrix} x_1^2 & x_1^3 & .. & x_1^m \\ x_2^2 & x_2^3 & .. & x_2^m \\ ... & .. & .. & .. \\ x_n^2 & x_n^3 & .. & x_n^m \end{bmatrix} && (6.7)\\
Y &= \begin{bmatrix} y_1 \\ y_2 \\ ... \\ y_n \end{bmatrix} && (6.8)
\end{aligned}
$$

Here, p_i is the i^{th} segment boundary.

Further, we hypothesize that the baseline is both autocorrelated and cross correlated. Hence, for the model order p, we model the regression as

$$
\begin{aligned}
y_i &= \alpha_1(\beta^1 x_i^1 + \beta^2 x_i^2 + .. + \beta^m x_i^m + \epsilon_i) \\
&+ \alpha_2(\beta^1 x_{i-1}^1 + \beta^2 x_{i-1}^2 + .. + \beta^m x_{i-1}^m + \epsilon_{i-2}) \\
&+ ... \\
&+ \alpha_{p+1}(\beta^1 x_{i-p}^1 + \beta^2 x_{i-p}^2 + .. + \beta^m x_{i-p}^m + \epsilon_{i-p}) && (6.9)\\
&= w_1 z_1 + w_2 z_2 + .. + w_m z_m + w_{m+1} z_{m+1} + .. + w_{(p+1).m} z_{(p+1).m} + \epsilon_i && (6.10)
\end{aligned}
$$

where α_a, β^b, w_c^b are coefficients, $z_{a.m+b} = x_b^a$, and $w_{a.m+b} = \alpha_a.\beta^b$. This results in a linear regression, where y_i is the target variable, z_i is the regressor, and $W = \{w_a | a = 1, 2..(p+1).m\}$ are the coefficients.

We want to learn the coefficients when we build the database. There are several methods addressing the learning problem. Ridge regression [54] and Support Vector Regression (SVR) [50] provide robust solutions to regression, especially when the data contain the outliers [5]. We use Ridge regression to obtain the coefficients $w_a^b = \alpha_a.\beta^b$.

We obtain the regression coefficients from,

$$W = (Z^T.Z + kI)^{-1}Z^TY \tag{6.11}$$

where I is an identity matrix. The positive Ridge regularizer k controls the conditioning of the problem by reducing the variance of the estimates [54].

We want to update the coefficients as we add new entries to the database. We use online gradient descent to achieve this [7]. The regularization function of the regression is

$$R(W) = \frac{k}{2}||W||^2 + ||W^TZ - Y||^2 \tag{6.12}$$

At time t, for update coefficient η, and regularization function $R(w)$ the standard gradient descent update is given by,

$$W_{t+1} = W_t - 2\eta(W^TZ_{t+1} - Y_{t+1})Z_{t+1} \tag{6.13}$$

Implicit update,

$$W_{t+1} = W_t - \frac{2\eta}{1 + 2\eta||Z_{t+1}||^2}(W^TZ_{t+1} - Y_{t+1})Z_{t+1} \tag{6.14}$$

modifies the learning rate, and makes the method more robust to scaling of the data [36].

6.3 Experimental Results

In this section, we present the experimental results for our method. We begin with the description of our dataset, and then present the results of the experiments.

In our experiments, we use the multi-parameter data from Challenge 2010 Test Set B at Physionet.org [21]. The database has 100 ECG waveform records, each contains six, seven, and eight-channel records of ECG, ABP and PPG signals, each 10 minutes long, to be used as a test set for the PhysioNet/Computers in Cardiology Challenge 2010. The recordings are digitized at 125 Hz.

We add AWGN to the first ECG channel of the data. We evaluate our method by quantifying the effectiveness of the reconstruction on this corrupted data. We compare the reconstructed data, and the corrupted data with the original data to evaluate the performance of our method. In the experiments, we compare the data with their baselines included. Therefore, the reconstructed signal is first obtained by merging the morphological reconstruction (Section 6.1), and the baseline reconstruction (Section 6.2).

We use the residual distance (Equation 5.9), and the disagreement (Equation 5.10) to evaluate our performance.

6.3.1 Experiment 1

We corrupt the first channel of the records with the additive white gaussian noise (AWGN) at 0dB SNR. We add the artificial corruption to only the last 20% of the data, and attempt to reconstruct it. We build the database from the first 80% of the data.

Table 6.1 lists the results of the experiment for the best 10 and the worst 10 records. In some records, e.g., record 37, the corruption, including total signal interruption, spans more than 50% of more than half of the signals in the record. Further, some records have signals misaligned with each other. Since we use correlated signals to fit and align the templates on the corrupted segment, this leads to large errors. Therefore, we report the results separately for 70 records that are relatively free of corruption and misalignment, and the rest.

Table 6.2 shows that our method performs well on records with enough correlated signals and correct alignment. Our method improves the signal quality on those records by reducing the average disagreement by 3300%, and the average residual distance by 280%.

Table 6.1: Experiment 1 : Effectiveness of data fusion on different sets of records

		In the record		In last 2 mins	Q1		Q2	
	Record ID	Beats #	PVC #	PVC #	$r_{S^b\#}$	r_{S^b*}	$\Delta_{S^b\#}$	Δ_{S^b*}
best 10 records	4	660	156	32	0.98	0.01	0.20	0.23
	18	744	1319	309	0.95	0.01	0.79	0.00
	23	852	0	0	1.04	0.01	0.09	0.00
	27	880	0	0	0.96	0.00	0.04	0.00
	58	809	0	0	0.99	0.01	0.01	0.00
	63	909	0	0	0.97	0.00	0.14	0.00
	78	729	0	0	0.97	0.00	0.25	0.09
	87	700	1	0	0.97	0.01	0.07	0.00
	93	671	515	147	0.98	0.00	0.74	0.12
	99	729	6	0	0.97	0.00	0.06	0.18
worst 10 records	24	848	28	21	0.92	0.41	0.17	0.16
	25	963	30	3	0.64	0.92	0.13	0.02
	33	1069	172	95	0.95	0.47	0.52	0.20
	37	837	0	0	0.74	1.24	0.02	0.05
	42	730	22	2	1.01	0.41	0.01	0.08
	45	1290	57	25	1.06	0.33	0.09	0.25
	55	814	9	0	1.12	0.53	0.04	0.00
	61	922	1	0	0.99	0.39	0.00	0.03
	75	775	89	22	1.44	0.85	0.18	0.21
	77	741	80	23	2.17	1.09	0.20	0.19

Table 6.2: Experiment 1 : Summary

	Records #		Q1		Q2	
			$r_{S^b\#}$	r_{S^b*}	$\Delta_{S^b\#}$	Δ_{S^b}
On good records	70	Mean	1.00	0.03	0.14	0.05
		Median	0.98	0.02	0.09	0.00
On poor records	30	Mean	1.27	0.33	0.13	0.11
		Median	0.96	0.22	0.09	0.05

6.3.2 Experiment 2

In this experiment we use only the 70 records that are relatively free of transient corruption and misalignment. We corrupt the first channel of the records with additive white gaussian noise (AWGN) at 0dB SNR. We add artificial corruption to the last 10% of the data. We also add the artificial corruption to a region as long as the 10% of the data starting at 20%, 40%, and 60% of the data. The idea is to keep the span of the corrupted region to 20% of the data, and evaluate the performance on identifying the corrupted regions, estimating the signal qualities correctly, and learning from the regions free of corruption. We build the database from the first 20% of the data, and add new entries when we come across a clean segment.

Table 6.3: Experiment 2 : Summary

Corruption starts at %		Q1		Q2	
		$r_{S^b\#}$	r_{S^b*}	$\Delta_{S^b\#}$	Δ_{S^b}
20	Mean	0.97	0.042	0.15	0.08
	Median	0.97	0.03	0.14	0.00
40	Mean	0.97	0.041	0.14	0.06
	Median	0.97	0.03	0.14	0.00
60	Mean	0.97	0.038	0.14	0.05
	Median	0.97	0.03	0.14	0.00

Table 6.3 presents the results of the experiment. Our framework's performance seems independent of the location of the corruption. However, there are two factors influencing the inferior performance compared to experiment 1. First, we have a smaller region to analyze (20% of the data, compared to 80% of the data). Second, our framework automatically identifies the corrupted region and reconstructs it. Another issue is that there are segments in the original record that are of poor signal quality, and our algorithm attempts to improve their signal quality. Our evaluation criteria is oblivious to this. It considers the original signal the gold standard and penalizes our method for cleaning up the original signal.

6.4 Summary

In this chapter, we presented a method for reconstructing a corrupted signal in a multi-parameter physiological signal by using information from correlated signals. The method uses database of templates for each pair of correlated signals, and finally combines the reconstructions from each correlated signal. We also predict the estimates of the baseline by treating them as a regression problem.

Chapter 7

Implementation

This chapter discusses the details of the implementation and analyzes the computational complexity of the system. Since our framework runs in real-time, we want to do the database operations quickly, and hence we provide methods to optimize these operations. We provide the details of the database construction in Section 7.1. In Section 7.2, we discuss the running time of each subcomponent and the over all computational complexity of our framework. Finally, in Section 7.3, we summarize our work.

7.1 Database

In our framework, we use a database of templates to provide the estimates for reconstruction. The database contains the corruption free segments learned, and the corresponding features. The use of a data structure helps us abstract the lookups, maintenance, and transfer of the morphological structures learnt.

A separate database is maintained for each pair of signals. This allows us to use the databases across the records. For instance, when we bootstrap, we begin with the database in the archive for the corresponding pair of signals, probably built from another patient's signals, and add the segments from the current record as we learn new morphologies. This allows us to transfer knowledge, and then adapt to the patient's signal morphologies.

7.1.1 Structure

Each database contains three fields: a set of features (Section 5.3.4), raw segments that are used to provide the estimates for reconstruction (Signal A), and the corresponding segments from the correlated signals (Signal B) as shown in Figure 7-1.

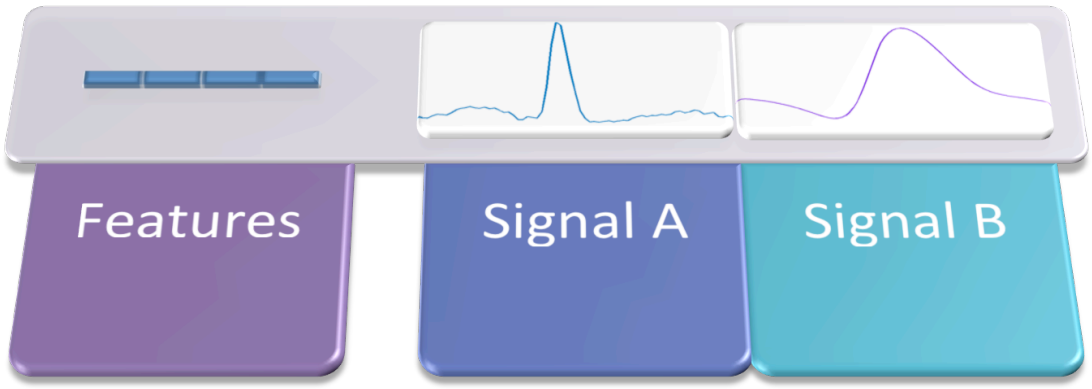


Figure 7-1: The structure of the database. It contains three fields: features, Signal A, and Signal B. The feature, itself, is a fixed length vector of dimension 30, and serves as the key.

7.1.2 Lookup

In reconstruction, to find the estimates for the corrupted segment, we lookup for an entry in the table that closely matches the current segment. We use the correlated signal (Signal A) to find the estimates of the signal that contains corruptions (Signal B). The entry contains the segments from both the signals. Since the lookup query and the entries in the table are of different lengths, we need to use an elastic distance measure, such as DTW. DTW takes $O(\ell^2)$ time, where ℓ is the length of the segment. The naive approach of comparing the query with every entry would take $O(n_d.\ell^2)$ time, for the database of n_d entries and segments of length ℓ. We use features to linearize this ($O(n_d.\ell)$) and provide a level of abstraction. The matching is first done through features, and then using DTW.

Our method uses the features to obtain the best 20 matches in the database. The lookup on this database is done using a nearest neighbor search. In nearest neighbor search, we find the entry closest to the query according to some distance function. We use Euclidean distance to compare two features, and hence, each dimension is weighed equally. To speed up the nearest neighbor searches, a multi-dimensional database is normally indexed by space partitioning or locality sensitive hashing (LSH). We use an algorithm that finds the approximate nearest neighbors in high dimensions. It has provably near-optimal time in the class of LSH [3].

The use of LSH adds another level of indirection. LSH indexes the data so that the queries for nearest neighbor can be served quickly. LSH is typically implemented by creating a hash table that puts similar items into the same bucket. To serve a query, we apply the hash function to the query, obtain the bucket's ID, and do an exhaustive search inside the bucket. The idea here is that the hash function returns a set smaller than the original data to search for. This set contains a smaller number of false positives, and may miss some positives. For any two vectors F_1 and F_2, our expectation is that

$$LSH(F_1) = LSH(F_2) \text{ if } ||F_1 - F_2|| < \delta \tag{7.1}$$

for some threshold δ. LSH also provides an indirect clustering of the data by grouping the similar items together.

7.1.3 Compaction

We begin with a database in an archive, and we add the entries to the database as we learn the morphologies. Since we want to limit the size of the database, we evict old, presumably less useful, entries from the database. This also helps us capture the most recent information well.

This is achieved by performing the compaction on the database when its size grows over a limit u_{upper}. We use Least Recently Used (LRU) algorithm to evict the old entries. During the compaction, the entries with the least counts are evicted, and the

size of the database is brought back to u_{lower}.

7.2 Computational Complexity

We expect to run the system in real-time. Therefore, the computational complexity of the framework is relevant.

- Segmentation : Segmentation uses DTW to compare a window and the template of a multi-parameter signal. If the length of the segment is ℓ, it takes $O(\ell^2)$ to find a segment boundary. If the signal length is n, the number of segments in the signal is $O(n)$. Hence, it takes $O(n.\ell^2)$ time to segment the signal.

- Signal Quality estimation : Similar to segmentation, the signal quality estimation uses an elastic function for comparison. However, it is done for all the signals in a multi-parameter signal, and hence, it takes $O(n.m.\ell^2)$ time for a multi-parameter signal with m parameters.

- Database Lookup : In Section 7.1.2, we showed that a lookup takes $O(\ell^2)$. The total number of segments requiring reconstruction is $O(n)$, and hence the total time complexity is $O(n.\ell^2)$

- Reconstruction : In reconstruction, we time-warp the template to fit the segment, and it takes $O(\ell^2)$ time for each reconstruction. Therefore, the total amount of time required is $O(n.\ell^2)$.

The total time complexity of our framework is $O(n.m.\ell^2)$, where n is the length of the signal, m is the number of parameters, and ℓ is the average segment length. Since m is typically a constant, the overall time complexity of our framework is $O(n\ell^2)$. Typically the sampling rate f of the signal determines the segment length ℓ. Hence, we can give the running time of the system on the sampling rate as $O(n.f^2)$.

We carried out our experiments using the unoptimized code written in Matlab 2010a, on a machine configured at 2.93GHz Intel Core i7 processor and 8GB Ram. The experiments (Sections 6.3.1, 6.3.2) conducted in Matlab R14, on 10 minute long

records from Challenge 2010 Test Set B at Physionet.org [21] took on average 51 seconds per record.

7.3 Summary

In this chapter, we discussed the details of the implementation of the system. First, we presented the description of the data structure that holds the templates, and the methods to access the data in the data structure. Second, we analyzed the computational complexity of the system. In the next chapter, we provide the summary and conclude.

Chapter 8

Conclusion and Future Work

We summarize our work in Section 8.1, review the major concepts and methods in Section 8.2, and outline some future research in Section 8.3.

8.1 Summary

We provided a set of novel computational tools to improve the signal quality of the physiological signals. We focused on: segmenting a multi-parameter quasiperiodic signal in the presence of noise and transient corruption, developing a signal quality estimate that is consistent across different signals, and reconstructing the segments of low signal quality. We formulated the problems, developed the models, and evaluated our solutions to the problems.

8.2 Contributions

We presented a novel method to jointly segment a quasiperiodic signal using template matching. The key idea of our work is that by simultaneously considering all the signals one can segment them more accurately than would be possible by considering each signal independently, especially in the presence of noise. Although we tested and applied our method only on physiological signals, the approach is generic and can be applied to any quasiperiodic signal. Template matching uses DTW, an elastic

distance function, to compare variable length sequences.

In order to extend the template matching to multi-dimensional signals we introduced weighted time warping (WTW) that allows the use of weights to vary the influence of each one-dimensional signal in the final alignment. We use the estimated quality of the signal for the weight. WTW measures the similarity between two multi-parameter sequences, even when the sequences are of different lengths.

We developed a new method, morphological dissimilarity, to estimate the quality of the signal at the granularity of a segment (heartbeat). We use morphological dissimilarity evaluated by an elastic distance function (LCSS) on the gradient of the signal to determine the signal quality. The signal quality Estimate obtained from the morphological dissimilarity is also comparable across different types of signals (ECG, ABP, and PPG).

We approached the problem of reconstruction as an unsupervised learning and discovery problem. We provided methods to learn and archive useful morphological features that can be used later in reconstruction. We provided methods to convert the variable length sequence into a fixed length feature vector.

We also demonstrated the effectiveness of our method. In the experiments carried out on the MIT-BIH Arrhythmia Database, a multi-parameter ECG database with many clinically significant arrhythmias, our method improved the classification accuracy of the beat type by more than 700% on signals corrupted with white Gaussian noise, and increased the similarity to the original signal, as measured by the normalized residual distance, by more than 250%. When the method was applied to the multi-parameter physiological signal data from Cinc Challenge 2010 database at Physionet.org, our method improved the classification accuracy of the beat type by more than 3300% on a signal corrupted with white Gaussian noise, and increased the similarity to the original signal by more than 280%.

8.3 Future Work

In the near future, we hope to evaluate the influence of the choice of design parameters on the performance of the system. These parameters include template update rate, length of the template database, signal quality estimate thresholds, etc. We set the parameters by locally optimizing the performance on MIT-BIH Arrhythmia dataset. First, we could like to find globally optimal values for these parameters through an exhaustive search on the same dataset. Second, we would like to test whether the same values give the best performance on another dataset, for example, on MIMIC II dataset.

We would also like to address a limitation of our present approach. Our method handles severe corruptions better than mild corruptions. The process of reconstruction is based on a binary decision, i.e., the method either decides to reconstruct a segment or chooses to leave it as it is. When the signal is mildly corrupted, for instance, at SNR 10dB as in Section 5.4.2, our method chooses not to reconstruct the signal. We would like to explore alternative approaches that work well on mild corruptions.

In our framework, we do not consider the temporal relationships between segments. Our hypothesis is that the correlated signals contain sufficient information to reconstruct the corrupted segment. Therefore, we do not map any temporally distant relationships between segments. We segment the signals to limit the error propagation, and hence, the temporal relationship between samples is limited to individual segments. We would like to investigate the effectiveness of incorporating temporal relationships between segments.

We hope to investigate the utility of transfer learning across patients. For instance, we would like to test the effectiveness of beginning the reconstruction process with a generic template database. We would then adapt to the current record by learning new morphological relationships in the current record. This is useful at the beginning of the reconstruction process when the database would be otherwise empty. The design of our framework supports this extension.

We tested our framework on ECG, ABP and PPG signals. We believe they are applicable to CVP and ICP as well. We hope to try our methods on other quasiperiodic multi-parameter signals to evaluate its usefulness.

In building our framework, we developed a set of tools, such as WTW, MD, and feature based indirection. We believe they will be useful in other areas as well. We would like to explore their use in areas such as video segmentations, multichannel audio segmentation, multi-parameter sensor data, etc.

8.4 Conclusion

The fundamental premise of this thesis is that given a multi-parameter signal, information in some signals can be used to infer the value of other signals. Based on this premise, we developed methods to jointly process a set of multiple signals, and improve their signal quality.

The key technical idea is to use templates to capture the relationships between the morphologies of correlated signals. These templates are learned on a patient-specific basis and continue to evolve as data is processed.

We showed that the joint segmentation could accurately segment a multi-parameter signal, even when each of the individual channels are so corrupted that they cannot be individually segmented. We also showed that correlated signals can be used to accurately reconstruct corrupted signals.

Our results strongly suggest that our initial premise is valid and that templates are indeed a good way to capture relationships among correlated signals. They also suggest that it is critical that the database of templates be allowed to evolve over time, perhaps because the signals themselves evolve over time.

Bibliography

[1] A Aboukhalil, L Nielsen, M Saeed, RG Mark, and GD Clifford. Reducing false alarm rates for critical arrhythmias using the arterial blood pressure waveform. *Journal of biomedical informatics*, 41(3):442–451, 2008.

[2] MG Akritas. Linear regression for astronomical data with measurement errors and intrinsic scatter. *arXiv.org*, 1996.

[3] A Andoni and P Indyk. *Near-optimal hashing algorithms for near neighbor problem in high dimensions*. Proceedings of the Symposium on Foundations of ..., 2006.

[4] L Arge, M de Berg, and HJ Haverkort. The priority R-tree: A practically efficient and worst-case optimal R-tree. In *Proceedings of the 2004 ...*, 2004.

[5] D Basak and S Pal. Support vector regression. *Neural Information Processing-...*, 2003.

[6] Hanqing Cao, Larry Eshelman, Nicolas Chbat, Larry Nielsen, Brian Gross, and Mohammed Saeed. Predicting ICU hemodynamic instability using continuous multiparameter trends. *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 3803–3806, 2008.

[7] N Cesa-Bianchi. Analysis of two gradient-based algorithms for on-line regression. In *Proceedings of the tenth annual conference on ...*, 1997.

[8] Samprit Chatterjee and Ali S Hadi. *Regression analysis by example*. LibreDigital, 2006.

[9] L Chen, MT Özsu, and V Oria. Robust and fast similarity search for moving object trajectories. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502, 2005.

[10] KL Cheung. Enhanced nearest neighbour search on the R-tree. *ACM SIGMOD Record*, 1998.

[11] G Clifford, F Azuaje, and P McSharry. Advanced methods and tools for ECG data analysis. *gbv.de*, 2006.

[12] Qiao Li Clifford and Gari D. Suppress False Arrhythmia Alarms of ICU Monitors Using Heart Rate Estimation Based on Combined Arterial Blood Pressure and Ecg Analysis. pages 1–3, mar 2008.

[13] AV Deshmane. False Arrhythmia Alarm Suppression Using ECG, ABP, and Photoplethysmogram. 2009.

[14] Elena Deza. Encyclopedia of Distances.

[15] H Ding, G Trajcevski, P Scheuermann, X Wang, and E Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

[16] AL Edwards. An introduction to linear regression and correlation, 1984.

[17] Z Fejzo. Adaptive Laguerre-lattice filters. *Signal Processing*, 2002.

[18] E Gil and J María Vergara. Detection of decreases in the amplitude fluctuation of pulse photoplethysmography signal as indication of obstructive sleep apnea syndrome in children. *Biomedical Signal Processing and ...*, 2008.

[19] A Gionis and P Indyk. Similarity search in high dimensions via hashing. In *Proceedings of the 25th International ...*, 1999.

[20] M Gleicher and L Kovar. Automated Extraction and Parameterization of Motions in Large Data Sets. *ACM transactions on graphics*, 2004.

[21] A L Goldberger, L A N Amaral, L Glass, J M Hausdorff, P Ch Ivanov, R G Mark, J E Mietus, G B Moody, C-K Peng, and H E Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation*, 101(23):e215–e220, 2000. Circulation Electronic Pages: http://circ.ahajournals.org/cgi/content/full/101/23/e215.

[22] J Guttag and Z Syed. Computational methods for physiological data. *dspace.mit.edu*, 2009.

[23] JD Hamilton. Time series analysis. Princeton Univ Pr, 1994.

[24] Patrick S Hamilton and Willis J Tompkins. Quantitative Investigation of QRS Detection Rules Using the MIT/BIH Arrhythmia Database. *Biomedical Engineering, IEEE Transactions on*, BME-33(12):1157–1165, 1986.

[25] A Hartmann. Reconstruction of missing cardiovascular signals using adaptive filtering. *Computing in Cardiology*, 2010.

[26] T Heldt, B Long, GC Verghese, P Szolovits, and RG Mark. Integrating Data, Models, and Reasoning in Critical Care. *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pages 350–353, 2006.

[27] WJ Hemmerle. An explicit solution for generalized ridge regression. *Technometrics*, 1975.

[28] Zhang Hengfei, Zeng Zhiyuan, Tan Xiaojun, and Chen Jixiong. 2010 2nd International Conference on Advanced Computer Control. In *2010 2nd International Conference on Advanced Computer Control*, pages 489–493. IEEE.

[29] A Henrich. The LSDh-tree: An access structure for feature vectors. *Data Engineering*, 1998.

[30] G Hinton. Learning multiple layers of representation. *Trends in cognitive sciences*, 2007.

[31] B Hjorth. The physical significance of time domain descriptors in EEG analysis. *Electroencephalography and Clinical Neurophysiology*, 1973.

[32] E Keogh and CA Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005.

[33] EJ Keogh and MJ Pazzani. Derivative dynamic time warping. *First SIAM international conference on data mining*, 2001.

[34] B-U Kohler, C Hennig, and R Orglmeister. The principles of software QRS detection. *Engineering in Medicine and Biology Magazine, IEEE*, 21(1):42–57, 2002.

[35] L Kovar and M Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics (TOG)*, 2004.

[36] B Kulis. Implicit online learning. *Proc Intl Conf Machine Learning*, 2010.

[37] LJ Latecki, V Megalooikonomou, Qiang Wang, R Lakaemper, CA Ratanamahatana, and E Keogh. Partial elastic matching of time series. *Data Mining, Fifth IEEE International Conference on*, pages 4 pp. EP–, 2005.

[38] Q Li, R Mark, and G Clifford. Robust heart rate estimation from multiple asynchronous noisy sources using signal quality indices and a Kalman filter. *Physiological measurement*, 2008.

[39] Y Li, Y Sun, P Sondhi, L Sha, and C Zhai. Reconstructing Missing Signals in Multi-Parameter Physiologic Data by Mining the Aligned Contextual Information.

[40] RG Mark and M Saeed. Temporal pattern recognition in multiparameter ICU data. 2007.

[41] GB Moody. The physionet/computing in cardiology challenge 2010: Mind the gap. *Computing in Cardiology*, pages 305–308, 2010.

[42] MD Morse and JM Patel. An efficient and accurate method for evaluating time series similarity. *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 569–580, 2007.

[43] C Myers, L Rabiner, and A Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(6):623–635, 1980.

[44] M Oefinger. Monitoring transient repolarization segment morphology deviations in Mouse ECG. *dspace.mit.edu*, 2006.

[45] Jiapu Pan and Willis J Tompkins. A Real-Time QRS Detection Algorithm. *Biomedical Engineering, IEEE Transactions on*, BME-32(3):230–236, 1985.

[46] Rao. Linear Statistical Inference and Its Applications, feb 1998.

[47] R Rodrigues. Filling in the Gap: a General Method using Neural Networks. *Computing in Cardiology*, 37, 2010.

[48] Mohammed Saeed and Roger Mark. A novel method for the efficient retrieval of similar multiparameter physiologic time series using wavelet-based symbolic representations. *AMIA Annual Symposium proceedings / AMIA Symposium AMIA Symposium*, pages 679–683, 2006.

[49] I Silva. PhysioNet 2010 Challenge: A Robust Multi-Channel Adaptive Filtering Approach to the Estimation of Physiological Recordings. *Computing in Cardiology*, 37, 2010.

[50] A Smola. A tutorial on support vector regression. *Statistics and computing*, 2004.

[51] JX Sun, AT Reisner, and RG Mark. A signal abnormality index for arterial blood pressure waveforms. *Computers in Cardiology, 2006*, pages 13–16, 2008.

[52] L Tarassenko, A Hann, and D Young. Integrated monitoring and analysis for early warning of patient deterioration. *Medical Applications of Signal Processing, 2005. The 3rd IEE International Seminar on (Ref. No. 2005-1119)*, pages 64–68, 2006.

[53] AN Tichonov. Solutions of ill-posed problems. Vh Winston, 1977.

[54] A N Tikhonov and Andreĭ Nikolaevich Tikhonov. *Ill-posed problems in natural sciences*. proceedings of the International Conference held in Moscow, August 19-25, 1991. Vsp, dec 1992.

[55] J Urrusti and W Tompkins. Performance evaluation of an ECG QRS complex detection algorithm. *PROC ANNU CONF ENG MED BIOL*, 1993.

[56] Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. Indexing Multidimensional Time-Series. *The VLDB Journal — The International Journal on Very Large Data Bases*, 15(1), 2006.

[57] J Wang. A new method for evaluating ECG signal quality for multi-lead arrhythmia analysis. *COMPUTERS IN CARDIOLOGY*, 2002.

[58] R Weber and HJ Schek. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the International ...*, 1998.

[59] SM Wu. Time Series and System Analysis Modeling and Applications, 1979.

[60] G UDNY YULE. *On a method of investigating periodicities in disturbed series, with special reference to Wolfer's sunspot numbers., Philos.* Transactions (A), 1927.

[61] Feng Zhou, F Torre, and JK Hodgins. Aligned Cluster Analysis for temporal segmentation of human motion. *Automatic Face & Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pages 1–7, 2008.

[62] W Zong and G Moody. A robust open-source algorithm to detect onset and duration of QRS complexes. *COMPUTERS IN CARDIOLOGY*, 2004.

[63] W Zong, G Moody, and R Mark. Reduction of false blood pressure alarms by use of electrocardiogram blood pressure relationships. *Computers in Cardiology 1999*, pages 305–308, 1999.